

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Final Report - Part II	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) VLSI Implementation of a Quantized Sinusoid Filter Algorithm and Its Use to Compute the Discrete Fourier Transform		5. TYPE OF REPORT & PERIOD COVERED 8/25/83 - 8/24/85
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Wiehrs Lewis Collier		8. CONTRACT OR GRANT NUMBER(s) N00014-83-K-0664
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering University of Rhode Island Kingston, RI 02881		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Electronics Program Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE March 1986
		13. NUMBER OF PAGES 99
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)  Approved for public release; distribution unlimited.		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Quantized sinusoids, DFT, VLSI, Narrowband Digital Filters		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The design, fabrication specification, and testing of custom Very Large Scale Integrated circuits which function as a chip set for realizing narrowband digital filters is described. The approach is based on a Quantized Sinusoid DFT algorithm. These integrated circuits use only a small VLSI chip area and provide fast operation. The design steps and testing are described in detail.		

LIBRARY  
RESEARCH REPORTS DIVISION  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA 93940

FINAL REPORT - PART. <sup>2</sup> ~~II~~

VLSI IMPLEMENTATION OF A QUANTIZED SINUSOID FILTER  
ALGORITHM AND ITS USE TO COMPUTE  
THE DISCRETE FOURIER TRANSFORM.

Wiehrs Lewis Collier

Department of Electrical Engineering  
~~The University of Rhode Island~~ *University.*  
Kingston, RI 02881

March 1986

Electronics Program  
Office of Naval Research  
Arlington, Virginia 22217

Under Contract N00014-83-K-0664

D.W. Tufts & G. Sadasiv, Principal Investigators

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

## **ABSTRACT**

The design, fabrication specification, and testing of custom Very Large Scale Integrated circuits which function as a chip set for realizing narrowband digital filters is described. The approach is based on a Quantized Sinusoid DFT algorithm. These integrated circuits use only a small VLSI chip area and provide fast operation. The design steps and testing are described in detail.

## CHAPTER 1

### INTRODUCTION

The availability of low cost, large density, high-speed Very Large Scale Integrated circuit (VLSI) devices has opened a new avenue for realizing increasingly sophisticated Digital Signal Processing (DSP) algorithms and systems [1]. The applicability of a VLSI implementation rests upon the fact that most DSP systems use only the arithmetic operations of addition and multiplication [2]. The regularity of such operations in VLSI allows for fast design times with standard cell libraries while imposing only a slight penalty in chip area due to the semi-custom nature of the system.

Despite the above advantages, a VLSI system can be inflexible. Custom processors may achieve analysis bandwidths of 10 - 50 MHz, but most are optimized for a specific application and would require extensive (and expensive) redesign to modify them to suit other applications [3]. This suggests that a VLSI design should contain some versatility. The integrated circuits described in this report have been implemented with this in mind. Although various uses for the IC system will be alluded to

throughout the report, its usefulness will be demonstrated by its use in reducing the time needed to calculate the Fourier coefficients [4] of a sequence.

This report describes the design, fabrication specification, and testing of a custom-VLSI based system for spectrum analysis and narrow band filtering. It is based on a quantized sinusoid algorithm which provides a simple but flexible method for calculating the values of the discrete Fourier transform (DFT) and for implementing a set of narrowband filters or beamformers. Chapter two examines the aspects of DFT calculations which will affect the proposed use of the design. Chapter three details the design of the Quantized Sinusoid DFT (QSDFT) algorithm. Chapter four contains specifications for the layout of the VLSI circuitry. Chapter five shows test results of the fabricated VLSI systems. Chapter six summarizes the research and suggests options for continued study.

## CHAPTER 2

### DISCUSSION OF DISCRETE FOURIER TRANSFORMS

Because of their importance, there has been much research regarding calculations of Fourier Transforms [5]. Following a brief description of discrete Fourier Transforms (DFT)'s, methods of reducing the computational requirements associated with DFT's will be discussed. (A more detailed analysis of the DFT may be found in many texts such as Digital Signal Processing [6].)

#### Definition of the DFT

For the special case where the data to be transformed is a finite sequence of real numbers,  $x(n)$ , with length  $N$ , the Fourier transformation pair is defined by the formulas

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad (2.1)$$

for  $k = 0, 1, 2, \dots, N-1$ , and

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad (2.2)$$

for  $n = 0, 1, 2, \dots, N-1$ , where

$$W_N = e^{-j(2\pi/N)}. \quad (2.3)$$

Equation (2.1) denotes the analysis transform and equation (2.2) represents the synthesis transform, also called the inverse DFT (IDFT). Because of the similarity of the two calculations, it shall suffice to perform only one of the transformation processes to determine the usefulness of any operation which promises a decreased DFT / IDFT calculation time. For the purposes of this report, only the analysis transform will be considered.

For a data sequence  $x(n)$  which is possibly complex in nature, the DFT  $X(k)$  of  $x(n)$  is given by

$$X(k) = \sum_{n=0}^{N-1} \{ (\text{Re}[x(n)]\text{Re}[W_N^{kn}] - \text{Im}[x(n)]\text{Im}[W_N^{kn}]) + j(\text{Re}[x(n)]\text{Im}[W_N^{kn}] + \text{Im}[x(n)]\text{Re}[W_N^{kn}]) \} \quad (2.4)$$

for  $k = 0, 1, \dots, N-1$ .

The computational requirements of this calculation are  $4N^2$  multiplies and  $N(4N-2)$  additions in the representation of the data ( e.g. reals or integers). For simplification, and for future hardware considerations, all arithmetic operations will be assumed to be integer operations.

### Reduction of DFT Calculation Requirements

As is seen from equation (2.4), the number of operations is proportional to  $N^2$ . A method to reduce the number of operations or the basic time required to perform each operation is needed.

Several algorithmic considerations have been introduced to reduce the number of calculations. These reduction methods are generally related to the length  $N$  of the sequence. One scheme which lowers the number of arithmetic operations is to calculate the Fourier transformation for a subset of the  $N$  frequencies given in equation (2.1). Algorithms such as the Goertzel [7] and the Chirp Z-transformation [8] calculate (with some limitations) the Fourier transform for only those desired frequencies,  $k$ 's, instead of all  $N$   $k$ 's. Other algorithms such as those by Runge [9,10], Danielson and Lanczos [11], and Cooley-Tukey [4] calculate all  $N$  of the  $X(k)$ 's but do so with only  $N \log_2 N$  operations instead of  $N^2$ .

These algorithms in which the goal is to reduce the total number of operations have been successful in conventional applications but may not be as robust in VLSI implementations. The order  $N \log_2 N$  algorithms, and the selective frequency algorithms reduce the number of calculations, and sometimes the storage requirements as well, but usually at the expense of added communication costs. For computer based applications, these algorithms are a necessity. For VLSI implementation, the extra communication costs can be detrimental. Fast VLSI multipliers can easily match and overrun standard integrated circuit input / output bandwidths. The silicon die size of these fast multipliers, though, does not allow for much extra circuitry on the same chip, thus necessitating DFT chip sets. Communication delays and arithmetic area have collided, creating the impetus for smaller processors - on chip to increase communication speeds, and with fast, possibly pipelined architectures.

#### Quantized Sinusoid DFT

One way to reduce both the operation time, and the area required for carrying out multiplications is to limit the class of the reference exponentials  $W_N$ . Runge and Konig [12] used the symmetry properties of sinusoids to

effectively reduce the number of calculations. The sinusoids may also be limited by restricting the allowable values of the sinusoid function.

By using approximated sinusoids with values which can be expressed as powers of two, (  $|\sin(x)| = 2^m$  and  $|\cos(x)| = 2^n$  ), the DFT multiply operation can be accomplished as a simple shift of the binary data representation. This multiply can be easily accomplished in hardware. VLSI technologies can perform this operation in a fraction of the area needed for conventional multiply circuitry. Pipeline architectures can actually hide the multiply time by converting it to communication time. The fast intrachip communication time further hastens the process.

The quantized sinusoid method, though, does have some disadvantages. Because the reference sinusoid is approximated, the calculated DFT is also an estimate. The quantized nature of the reference sinusoids also imparts spurious sidelobes into the DFT filter's response [13].

The quantized sinusoid DFT offers advantages not found in other algorithms. The reduced silicon area and improved operation speed have already been alluded to. Efficient decoding of the sinusoids allows for the selection of the desired transform frequencies, and also allows for almost

infinite precision in the specification of the desired frequencies rather than the  $1/N$  range rendered by equation (2.3). This DFT offers a quick estimate of the principle components of a signal which may be used in frequency estimation algorithms such as those proposed by Tufts and Melissinos [14].

### Summary

DFT's for finite length sequences may be calculated as shown in equations (2.1) and (2.2). Conventional techniques require  $N^2$ , or at best  $N\log_2 N$ , lengthy calculations. The quantized sinusoid DFT utilizes a reduction in the number of operations, AND, a reduction in the time taken to perform the associated arithmetic operations to provide a fast estimate of the DFT of a sequence. The hardware also has uses as part of a general purpose filter.

## CHAPTER 3

### QUANTIZED SINUSOID DFT ALGORITHM

Given the applicability of the quantized sinusoid method to calculate a discrete Fourier transform, the implementation of the algorithm must be considered. It will be assumed that the ultimate goal is to realize the system via silicon fabrication. The remaining algorithm parameters must now be decided upon for the implementation of a VLSI circuit. Once these parameters for the quantized sinusoid DFT algorithm have been described, they will be combined to specify the complete algorithm.

#### Quantization of Sinusoids

The following constraints exist for the sinusoid quantization method which is to be used. Foremost, the quantized sinusoid should emulate a sinusoidal wave as well as possible. One may use the level of spurious sidelobes introduced by the quantized sinusoids when they are used in a filter implementation as a measure of the appropriateness for the chosen quantization [13]. The other restriction is that the quantization levels be powers of 2 to simplify the hardware. Thus, the first half cycle of the desired

quantized sinusoid will take the form of that shown in Figure 3-1.

For the sinusoid given in Figure 3-1, only the number of levels and the transition points between the levels need to be determined. It has been shown [13] that a sinusoid such as is shown in Figure 3-2 is, in fact, a good approximation to non-discrete sinusoids. The spurious sidelobes are 24.17 dB below the signal response of the filter in which the quantized sinusoids are used. The small number of quantization levels reduces the hardware complexity. The transition points are at points which are also roughly powers of two which will simplify the decoding of the sinusoids. Larger numbers of levels were not used since they do not provide significantly larger sidelobe reduction [13].

#### Decoding of the Sinusoids

Once the form of the quantized sinusoids has been determined, a method of decoding the appropriate phase must be determined. It shall be assumed that there are  $p$  bits in which the phase of an entire cycle of the quantized sinusoid may be expressed. The transition points shall be made to coincide with powers of two such that the first cycle of a quantized sine wave appears as shown in Figure 3-3. Use of

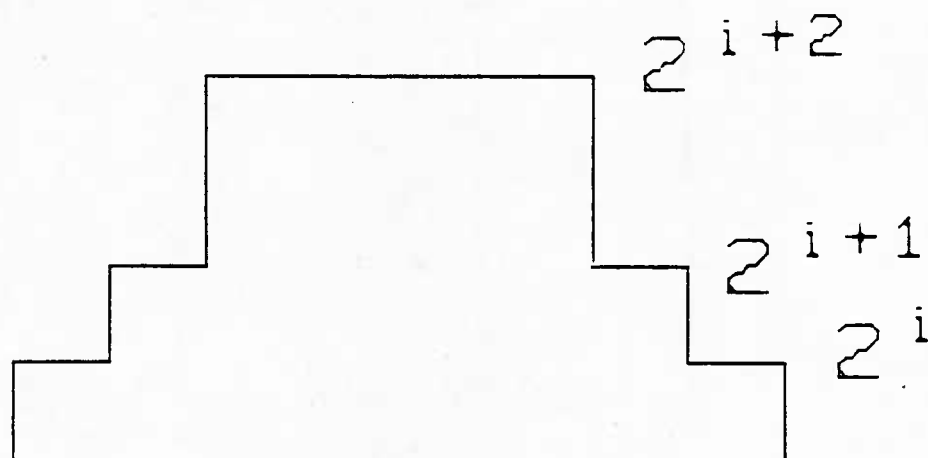


Figure 3-1  
One Half Cycle of a Quantized Sinusoid.

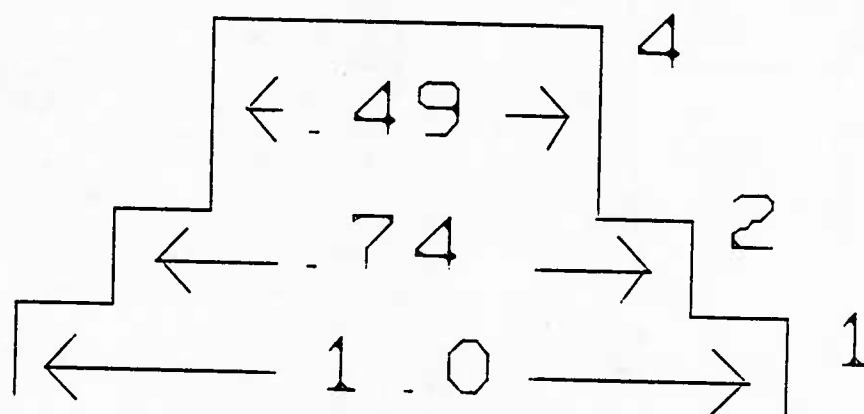


Figure 3-2  
Detailed Half Cycle of a Quantized Sinusoid

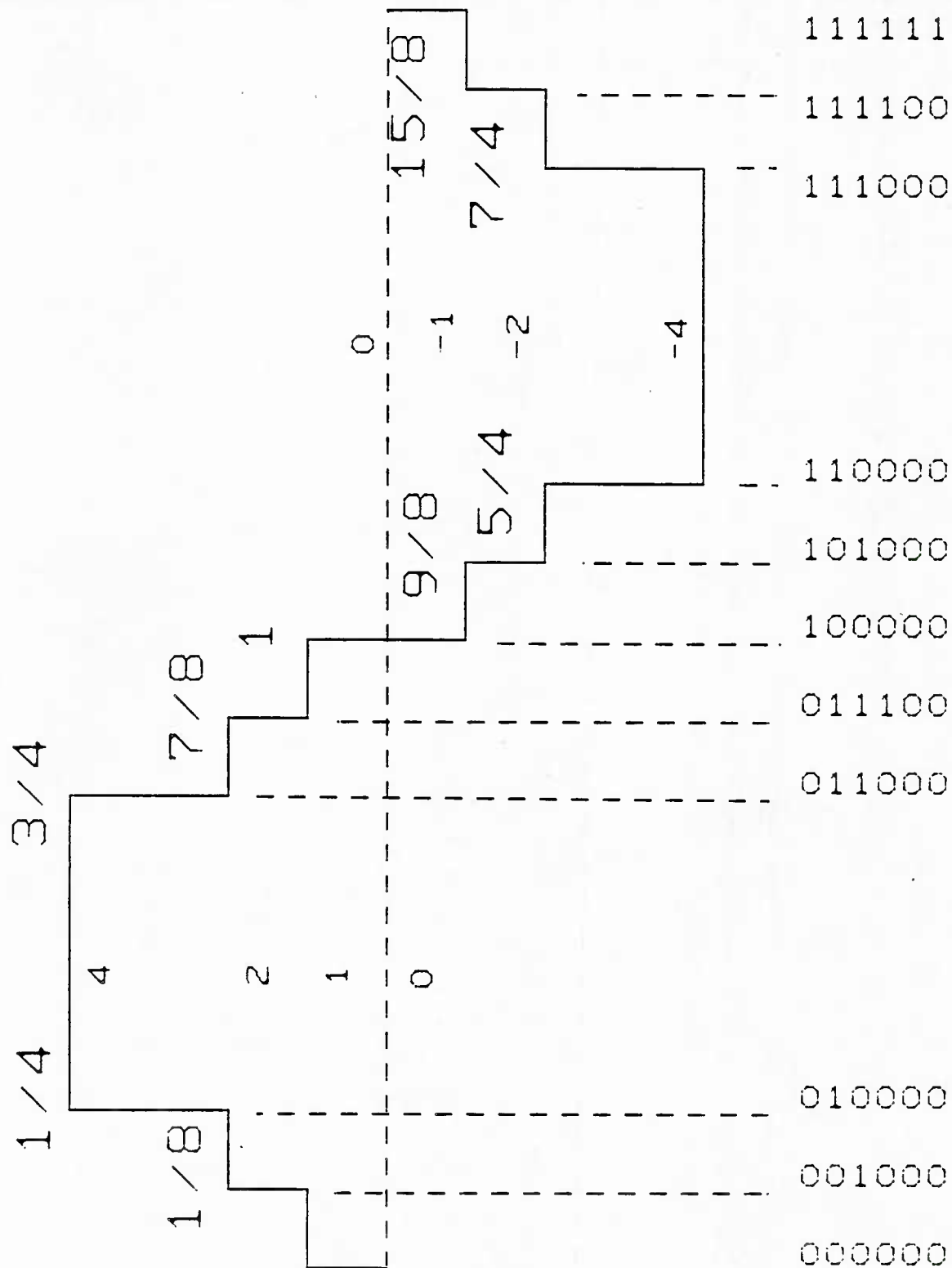


Figure 3-3

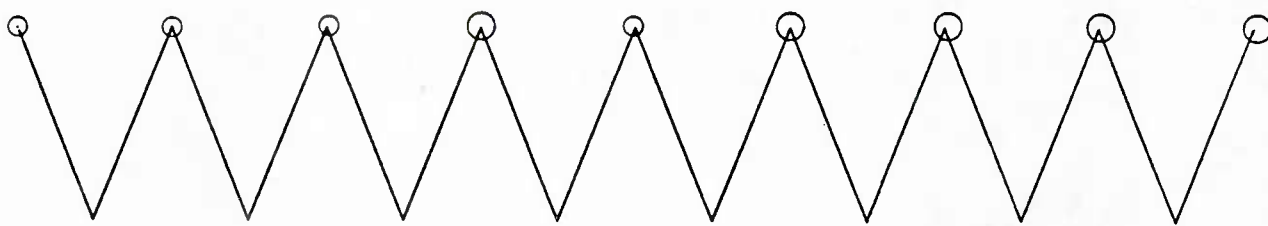
Full Cycle of Sinusoid with Decoding Information

this will raise the sidelobes to 23.52 dB below the fundamental [13].

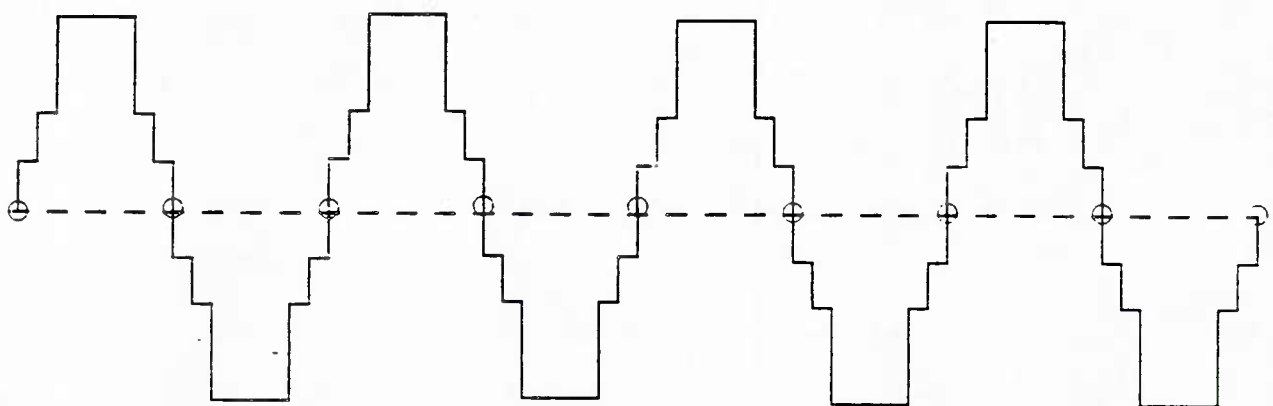
The reference waveforms will depend upon the data sampling rate as shown in Figures 3-3 and 3-4. The  $p$  bits allow for a large range of values for the frequencies of the reference sinusoids. As is seen, the maximum reference frequency which is allowed is  $(p-1)/p$  times the sampling rate. In practice, this high a reference frequency will never be used so that this upper limit on reference frequencies will not be a restriction.

The reference sinusoids, then, will be expressed as phase increments based upon the sampling rate of the input data. For example, as seen in Figure 3-3, a phase increment of 111...11 would represent a frequency one bit in  $p$  less than the sampling rate. A phase increment of 000...01 equates to a reference frequency of the sampling rate divided by  $2^p$ .

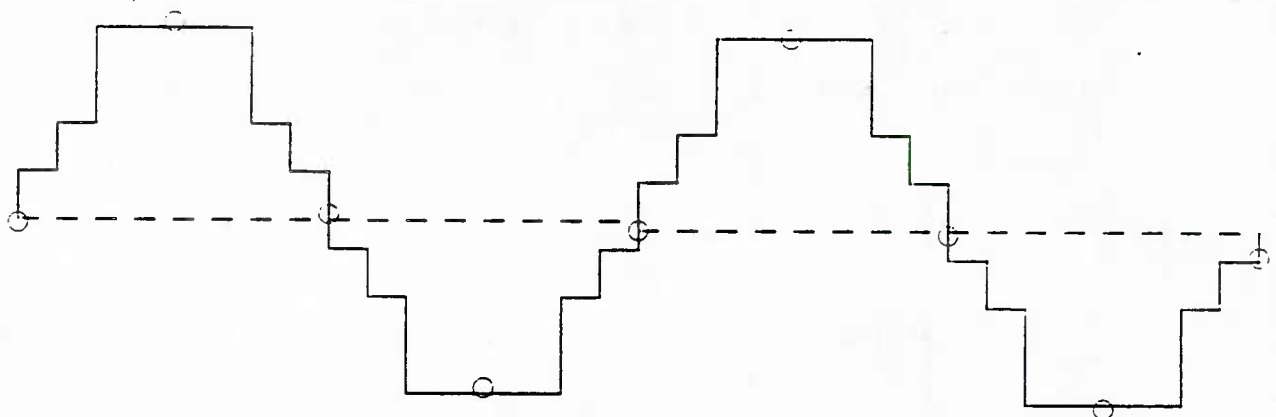
By utilizing the symmetric property of the proposed quantized sinusoid, a decoding scheme has been devised to quickly determine the correct phase of the reference sinusoids. If, for instance, we wish to follow the paths of the reference frequencies equal to one half and one quarter the sampling rate, i.e.  $R_p = 100...00$  and  $010...000$



Sample freq  $F_s$



$$R_p = 100000 = F_s / 2$$



$$R_p = 010000 = F_s / 4$$

Figure 3-4

Relationship Between the Reference Frequencies and  $R_p$

respectively, the positioning of the reference phases along the quantized phase is shown in Figure 3-4. The phase register  $R_p$  can be accumulated to walk along the reference wave of the DFT.

The symmetry of the quantized wave allows the four quarters of the wave to be considered as one section, traversed four times during each cycle. A half cycle for the case  $p = 5$  is shown in Figure 3-5. Walking up one side of a sine half cycle and back down the other back to the zero level requires only two bits of decoding. A third bit tracks the direction of the walk, (i.e. either up or down), and a fourth bit is the sign bit. This is easily extended to as many  $p$  bits as needed since only the four most significant bits are encoded. Thus, the reference frequencies have infinite precision limited by the hardware, not  $N$ .

The decoding of the sinusoid wave can be achieved by using a simple Programmable Logic Array (PLA), a lookup table or random logic. The decoding for the reference cosine wave is similar but, of course, is shifted by 90 degrees. The decoding scheme is given below in Table 3-1. The sign bit is not decoded here, but is saved for use with the incoming data to determine the effective sign of the quantized arithmetic multiplication.

Count Direction  
Set To 1

Count Direction  
Set To 0

	0 0 0 0 0	
	0 0 0 0 1	0 0
	0 0 0 1 0	
	0 0 0 1 1	Decodes as
0 0	0 0 1 0 0	1st Level
	0 0 1 0 1	
	0 0 1 1 0	
Decodes as	0 0 1 1 1	7
3rd Level		
	0 1 0 0 0	
0 1	0 1 0 0 1	0 1
	0 1 0 1 0	
	0 1 0 1 1	Decodes as
	0 1 1 0 0	2nd Level
	0 1 1 0 1	
	0 1 1 1 0	
15	0 1 1 1 1	15
	1 0 0 0 0	
1 0	1 0 0 0 1	
	1 0 0 1 0	
Decodes as	1 0 0 1 1	
2nd Level	1 0 1 0 0	
	1 0 1 0 1	1 0
	1 0 1 1 0	
23	1 0 1 1 1	Decodes as
		3rd Level
	1 1 0 0 0	
1 1	1 1 0 0 1	1 1
	1 1 0 1 0	
Decodes as	1 1 0 1 1	
1st Level	1 1 1 0 0	
	1 1 1 0 1	
	1 1 1 1 0	
	1 1 1 1 1	

Figure 3-5

Phase Increment Accumulation Decoding With 5 Bits.

---

Inputs from p-2, p-3,  
& p-4 bits of the  
phase accumulator

Sine Decode

Cosine Decode

0	0	0	0	0	1	1	0	0
0	0	1	0	1	0	1	0	0
0	1	0	1	0	0	0	1	0
0	1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0	1
1	1	0	1	0	0	0	1	0
1	0	1	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0

TABLE 3-1

Decoding of Quantized Sinusoids.

---

### Data Representation

The input form of the data to be processed must be compatible to the quantized sinusoid DFT algorithm. Since the multiplies will be evaluated by shifts of the binary data, its representation must be such that a shift by  $i$  places is equivalent to multiplication by  $2^i$ . Several of the most popular data representations will now be considered.

The first choice is to represent the data in a sign/magnitude format. In this representation, the value of

the number is derived from its binary digits by Equation (3.1) for the N bit representation.

$$\text{value} = (-1)^{b_{N-1}} \left\{ \sum_{j=0}^{N-2} b_j 2^j \right\} \quad (3.1)$$

Shifting by one place creates an N + 1 bit representation so the shifted value is given by Equation (3.2).

$$\text{value} = (-1)^{b'_N} \left\{ \sum_{k=0}^{N-1} b'_k 2^k \right\} \quad (3.2)$$

It is assumed that a zero is shifted into  $b'_0$ . Also, the  $(-1)$  term affects only the sign of the value, and may be cancelled when comparing Equations (3.1) and (3.2). Since  $b'_0 = 0$  by definition of the performed shift, Equations (3.1) and (3.2) can be used to show (Equation 3.3) that the shift of a sign magnitude binary number is, in fact, the same as multiplication by two.

$$\text{SHIFT} \left[ \sum_{j=0}^{N-2} b_j 2^j \right] =$$

$$\begin{aligned}
&= \left[ \sum_{k=1}^{N-1} b'_k 2^k \right] \\
&= 2 * \left[ \sum_{l=0}^{N-2} b_l 2^l \right] \quad (3.3)
\end{aligned}$$

The two's complement representation may also be used in the quantized sinusoid DFT algorithm. In this format, the value of an N bit word is

$$\text{value} = -2^{N-1} + \left\{ \sum_{j=0}^{N-2} b_j 2^j \right\}. \quad (3.4)$$

Shifting by one place gives the N + 1 bit value

$$\text{value} = -2^N + \left\{ \sum_{k=0}^{N-1} b'_k 2^k \right\}. \quad (3.5)$$

Again  $b'_0 = 0$  so that

$$\begin{aligned}
\text{SHIFT} & \left[ -2^{N-1} + \sum_{j=0}^{N-2} b_j 2^j \right] \\
&= \left[ -2^N + \sum_{k=1}^{N-1} b'_k 2^k \right] \\
&= 2 * \left[ -2^{N-1} + \sum_{k=1}^{N-1} b'_k 2^{k-1} \right] \\
&= 2 * \left[ -2^{N-1} + \sum_{l=0}^{N-2} b'_l 2^l \right]. \quad (3.6)
\end{aligned}$$

As with the sign magnitude representation, a shift of the binary data is equivalent to multiplication by two.

A popular representation which may not be used with the previously defined shift is the one's complement format. To see why the one's complement representation does not work, one needs only to look at the representation of a negative

number. A -1, when shifted, becomes -3, not the desired -2. A remedy for this is to shift in a '1', instead of a '0'. Extra control circuitry can be easily constructed which will allow for the use of the one's complement representation.

There are, of course, some remaining binary codes which may behave properly when shifted. Since most digital output devices use one of the previously mentioned codes, no time will be invested to ascertain if one of these other codes might, in fact, be useable with only moderate alteration of the hardware.

#### Algorithm Operation

The operational methods by which the Quantized Sinusoid DFT may be calculated will now be discussed. Two algorithms will be considered, one in which the input data will be stored for further processing, and one in which the partially formed output coefficients will be stored. It will be assumed that a length  $N$  data sequence will be processed and that there are  $M$  reference frequencies.

The Coefficient Storage (CS) method for calculating the DFT operates as follows. For each input data value,  $x(n_i)$ , the phase of each associated reference sinusoid frequency  $k_j$ , for  $j = 1$  to  $M$ , is calculated and stored. This phase

value is in turn decoded to determine the correct shift of the input data to emulate the multiplication by the reference frequency. The current Fourier coefficient  $X(k_j)$  is accumulated and stored. The next coefficient  $X(k_{j+1})$  is then accumulated in a similar manner. This continues until all  $M$  coefficients have been computed for the given data value  $x(n_i)$ . This cycle repeats  $N$  times for the data segment length as is shown in Figure 3-6. At this time, all  $M$  Fourier coefficients are complete. A data flow path diagram of the CS method is shown in Figure 3-7.

The Data Storage (DS) method works similarly to the CS method except that the operation order is reversed. A new reference frequency  $k_i$  is input. For each of  $N$  (which is constrained to be equal to  $M$ ) data points, the phase is calculated, the shift determined, and the coefficient is updated. After  $N$  data values, the final coefficient  $X(k_i)$  is then complete. The coefficient for the next reference frequency  $k_{i+1}$  is then similarly computed as outlined in Figure 3-8. The data flow diagram for the DS method is given in Figure 3-9.

The Data Storage method has several advantages over the Coefficient Storage method. A large amount of silicon area is saved due to a reduced storage demand. The phase

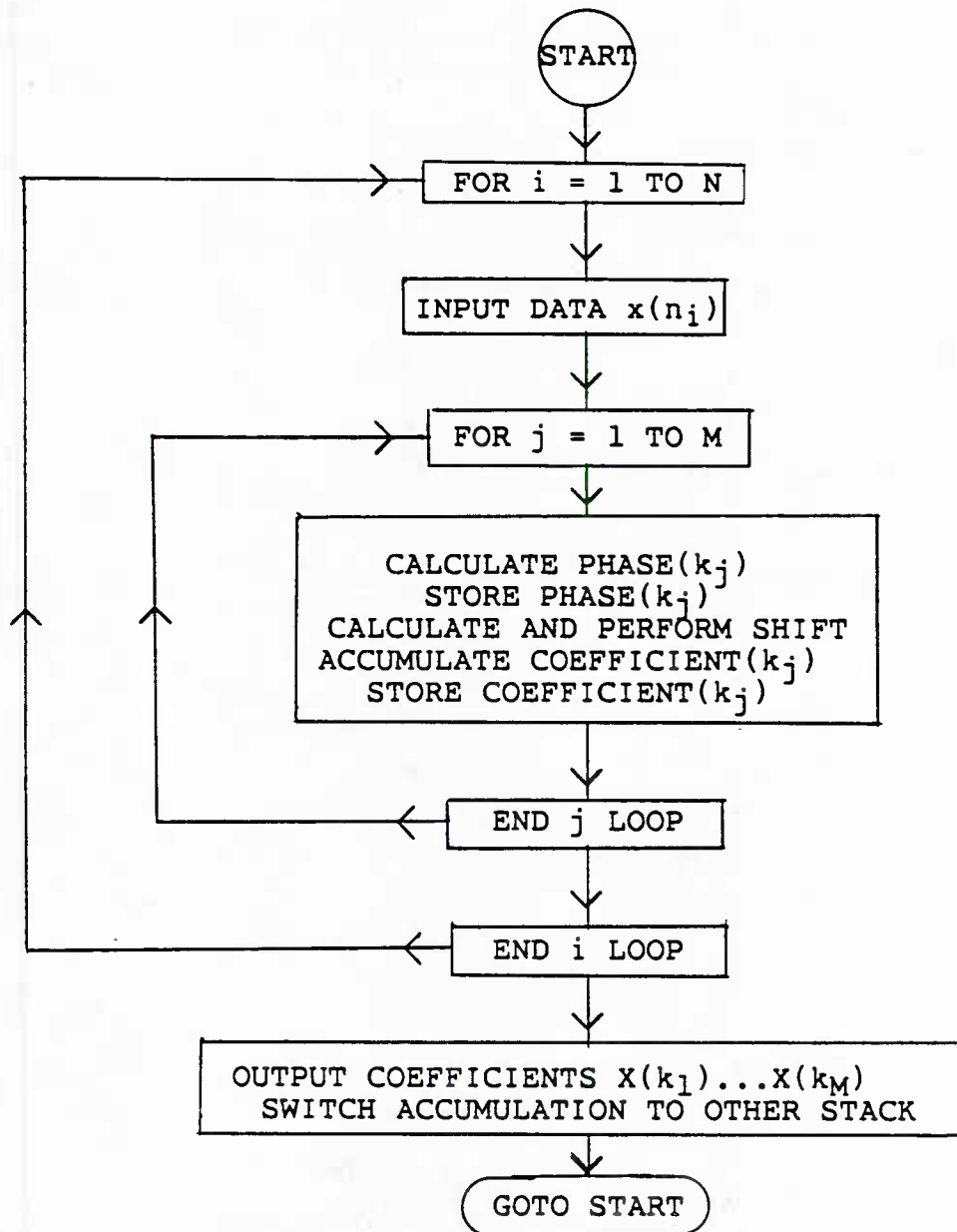


Figure 3-6  
CS Method Operation Flowchart.

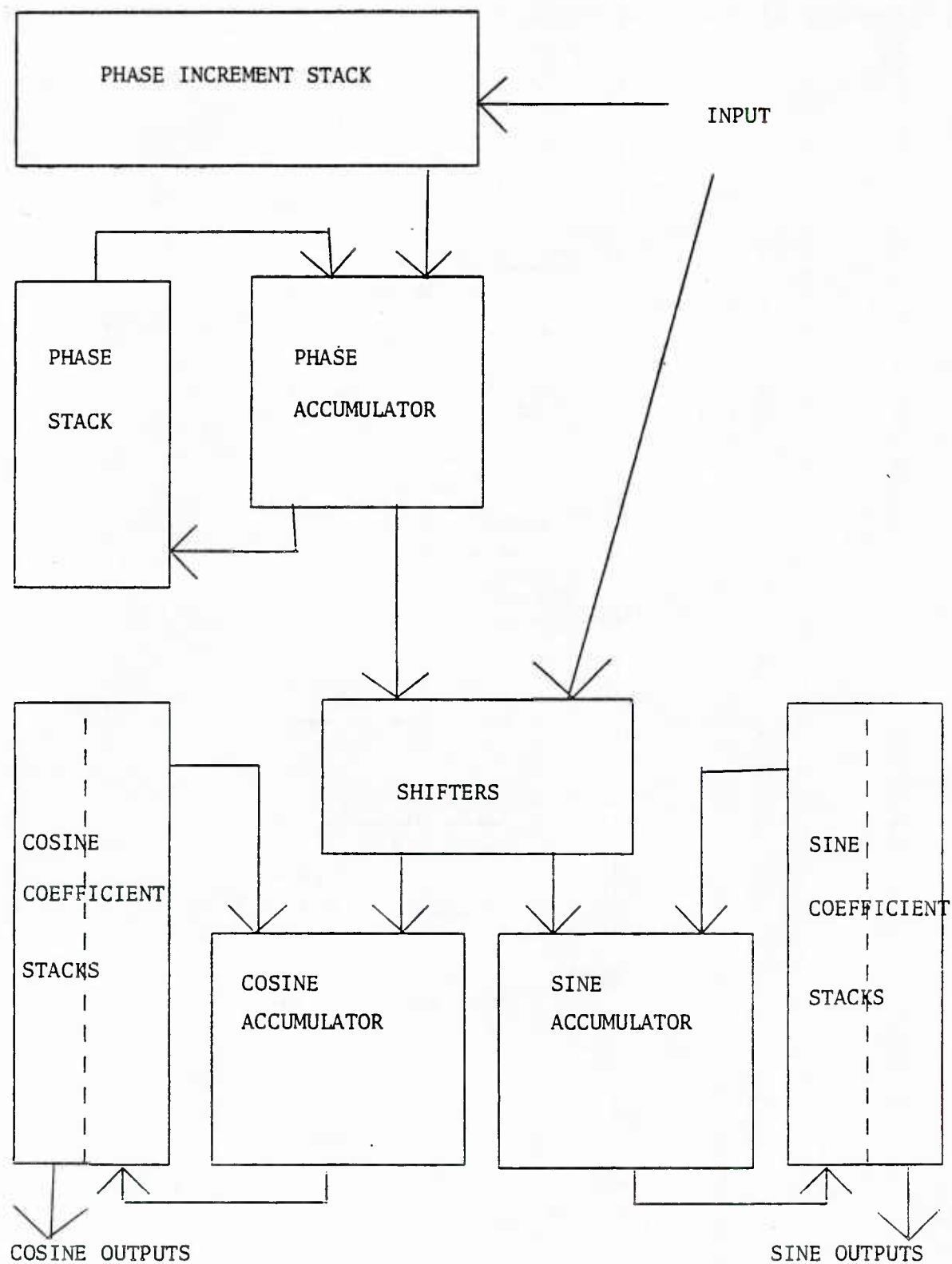


Figure 3-7

CS Method Data Flow Paths

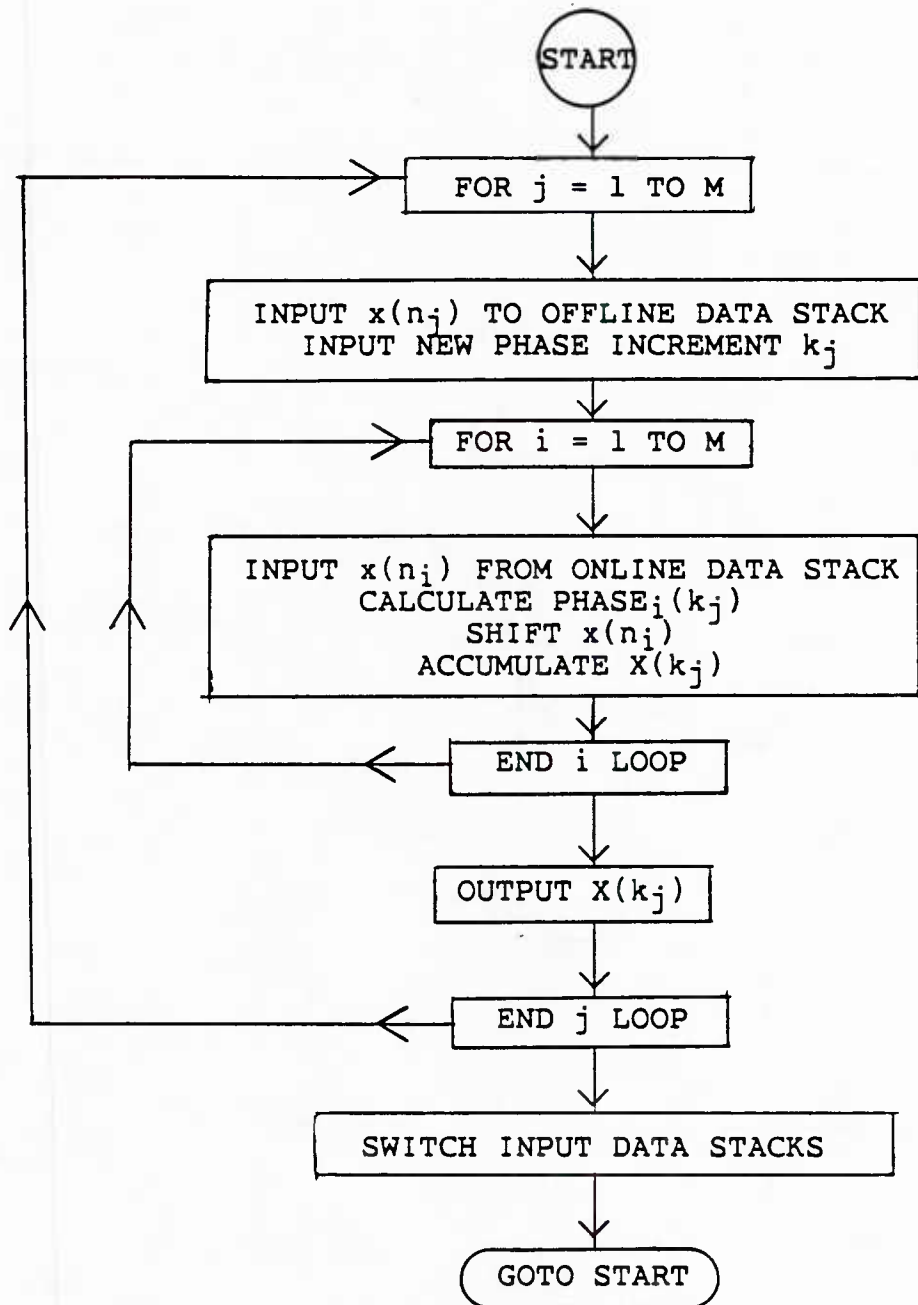


Figure 3-8  
DS Method Operation Flowchart.

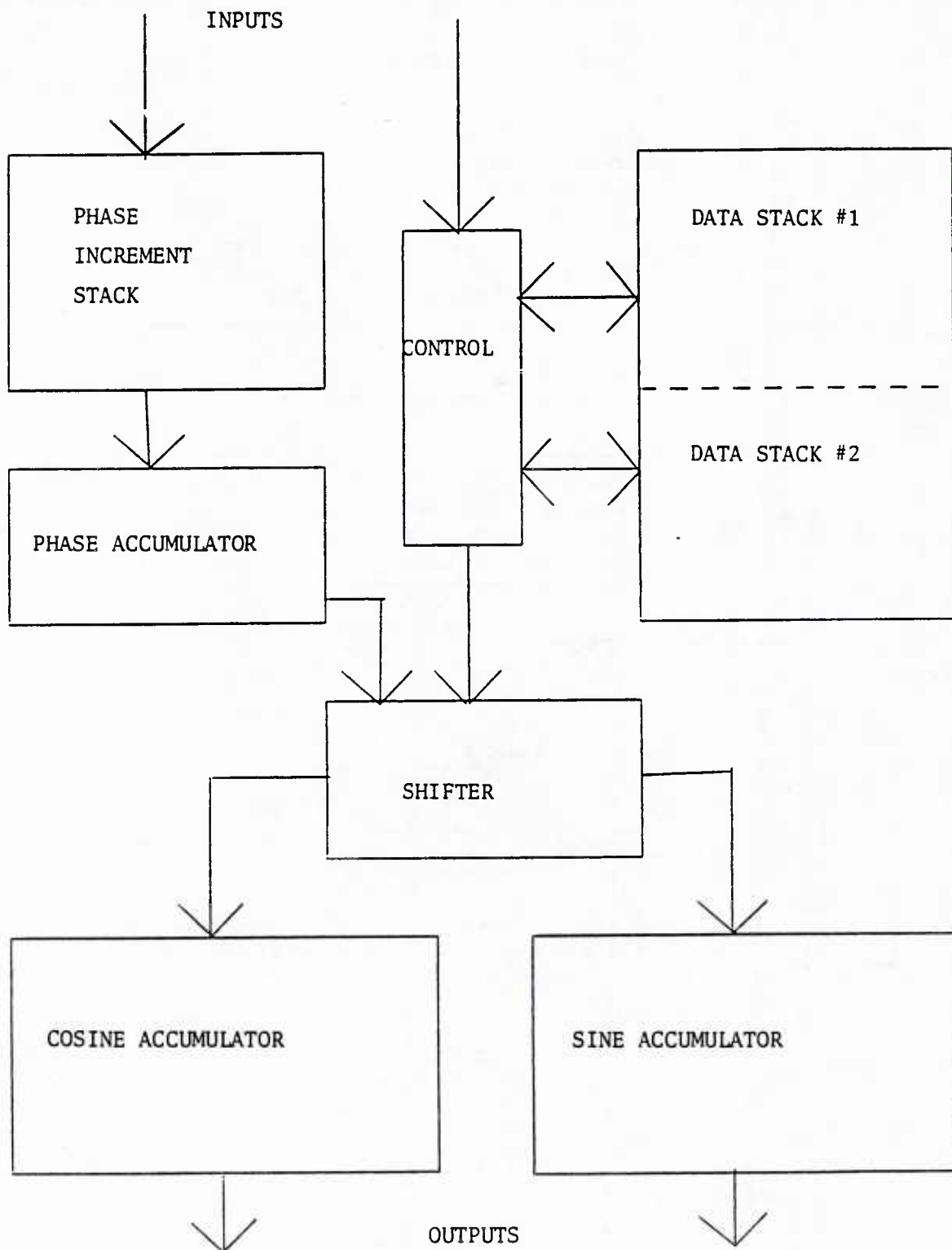


Figure 3-9  
DS Method Data Flow Paths

increment and data storage stacks are considerably smaller in the DS method than are the coefficient, phase increment, and phase storage stacks in the CS method. This is because the summed coefficients are inherently larger than the data itself. Also, the extra storage elements for the incremental phases are eliminated in the DS method. Another advantage of the DS method is that the coefficients may be removed periodically, instead of all at once as in the CS method. This reduces input/output problems by allowing for serial output which decreases pin count and necessary I/O bandwidth. This output scheme is also more useful in applications where further processing upon the coefficients is desired such as is the case when the hardware is used as a low pass filter [13] or to generate a starting vector for effective computation of principle eigenvectors [14].

The large amount of data movement is the source of the major disadvantage in the CS method. In addition to the increased storage requirements which were alluded to previously, there is an increased amount of communication within the chip or chip set. Such a penalty would result in an increased number of large width data busses needed to place in parallel all the data where it should be. These added costs inhibit effective pipelining. These restrictions will also increase the amount of required control circuits.

The major disadvantage of the DS method is the relationship of  $N$  to  $M$ . This dependence results from the storage of data in the DS method. As the computation of each coefficient  $X(k)$  concludes, a new data value is acquired. Since there are  $M$   $k$ 's, there must be  $M$   $x(n)$ 's, and conversely. Data sequence lengths which are larger than the reference frequency count may be transformed by padding the frequency list with don't care frequencies. (This, of course, will reduce the overall effectiveness of the QSDFT.) Another way in which this DS handicap may be overcome is to use several DS method QSDFT chips in parallel to segment the incoming data stream. This would effectively cause several short DFT's of length  $M$  (where  $M < N$ ) to be calculated instead of a single length  $N$  DFT. 'Normal' DFT procedures such as overlap and add could then be used to combine the DFT sequence.

#### Final Quantized DFT Algorithm

A finalized implementation may now be gleaned from the previous discussion of important algorithm parameters. The major subsystems of the design will be discussed, along with their interactions. The operation of the system, along with the plans for testing will be considered.

The quantized sinusoid to be used as the reference wave was shown in Figure 3-3. 10 bits shall be used to represent a complete cycle. Input will be 8 bits of sign/magnitude or two's complement data. The phase will be decoded using a PLA. The PLA output will be directed towards a shifting network which will displace the data. The calculations will be made using the Data Storage (DS) method. Initially, the algorithm will be implemented as a 3 chip set, with each of the following major subsystems residing on a separate chip.

The first subsystem is the shifting network. For our purposes, a simple multiplexing network may be used. This offers a fast method of selecting the properly shifted version of the input data. The multiplexor inputs are driven by buffered outputs from the decoding PLA. The PLA inputs are the three output bits from the phase accumulator  $O_{N-2}$ ,  $O_{N-3}$ , and  $O_{N-4}$  as seen in Figure 3-10.

The phase accumulator itself is the second subsystem. It will take as inputs the phase increments of 10 bits. It also uses the system reset and control lines as shown in Figure 3-11. The adder unit consists of a ripple carry adder and circuitry for storing the input phase increment and the resultant output. A method for serially shifting the adder output is also provided to facilitate testing.

4 MSB From Phase Increment  
Accumulator

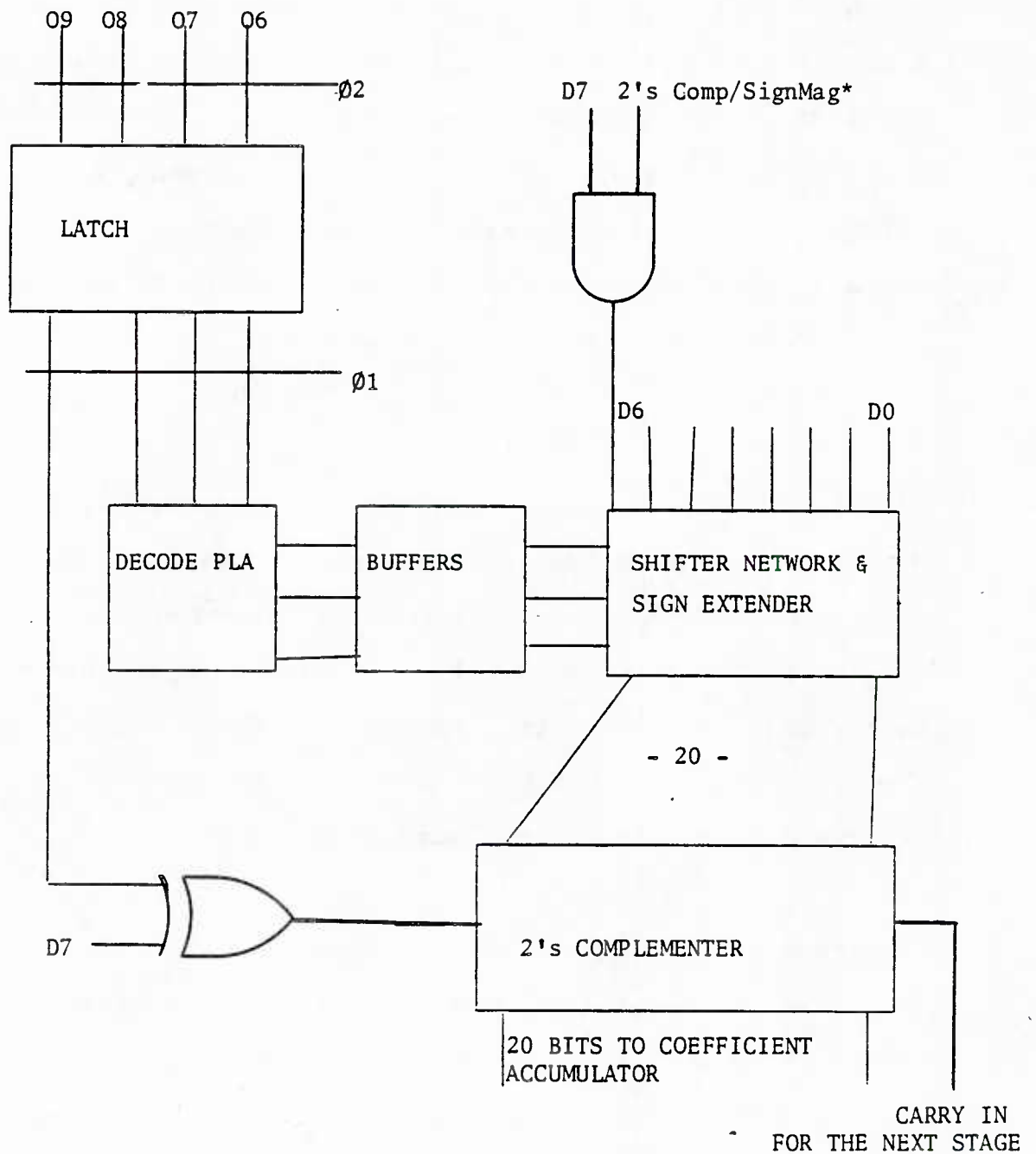


Figure 3-10  
Shifter Network Block Diagram.

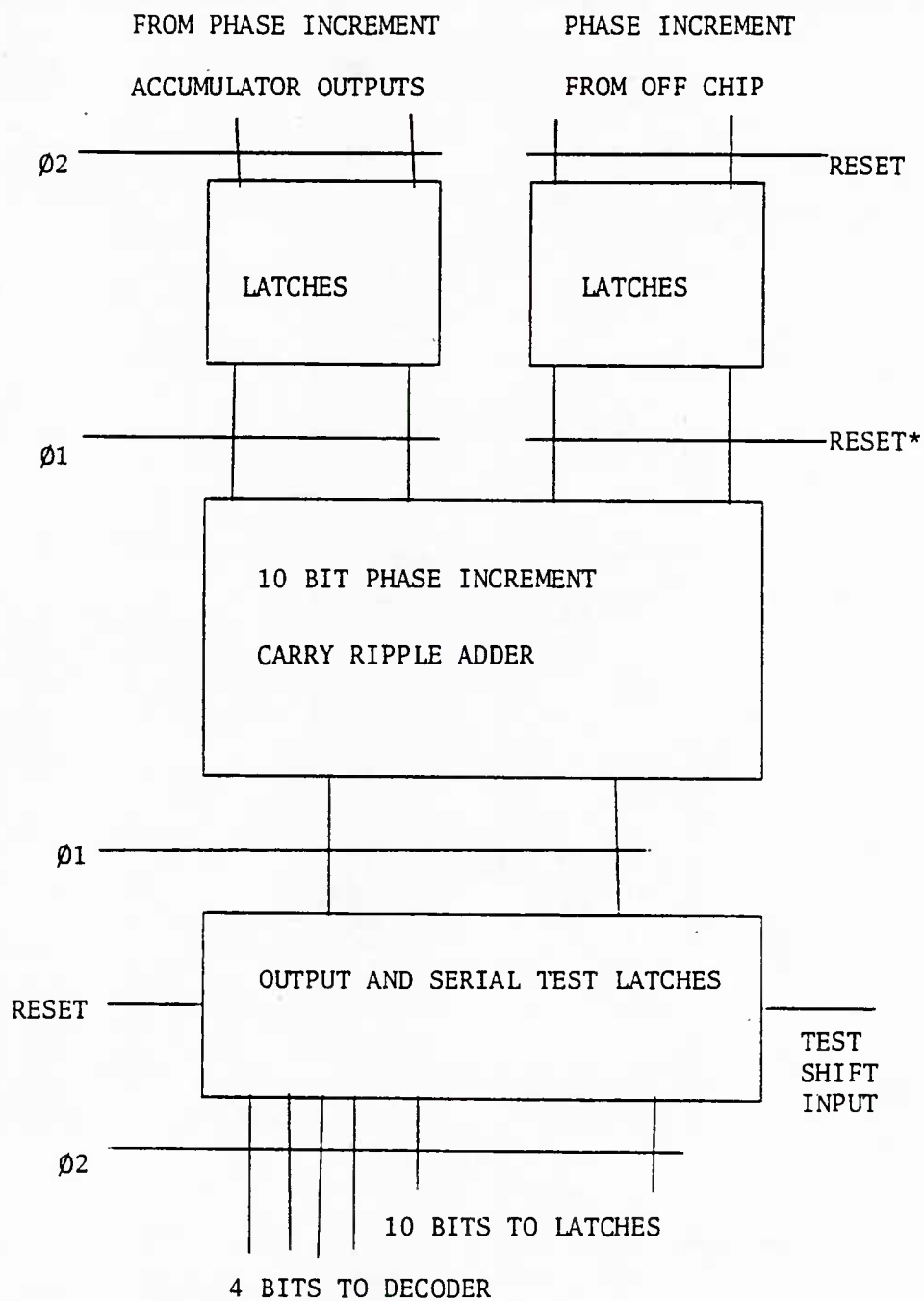


Figure 3-11  
Phase Increment Accumulator Block Diagram.

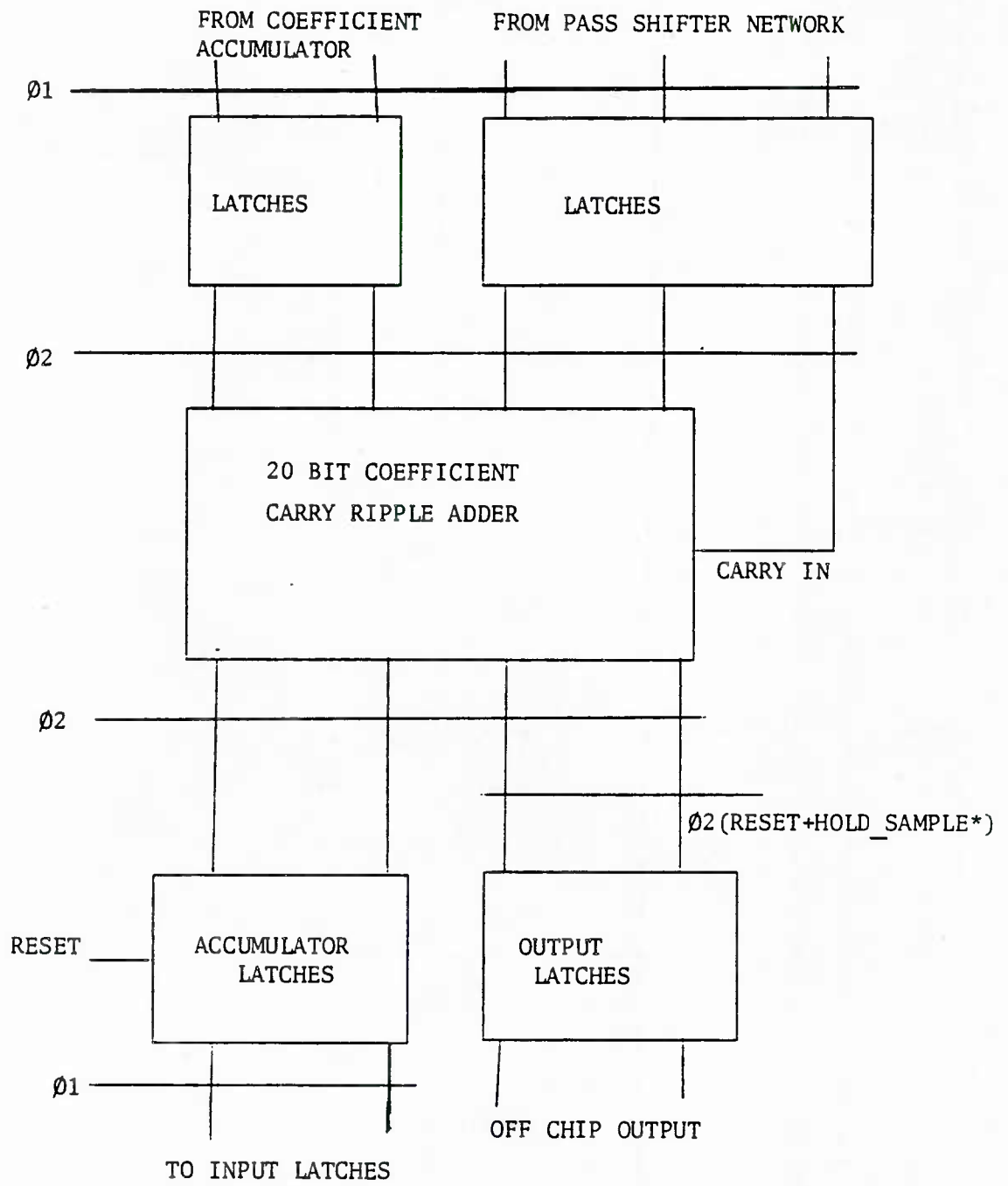


Figure 3-12  
Coefficient Accumulator Block Diagram.

The final section is the coefficient accumulator. It uses the same basic adder unit as is in the phase accumulator. The coefficient accumulator is 20 bits wide to allow for data sequences up to length 1024. Also included in this subsystem is the control circuitry. In this logic are circuits to generate resets based on external user defined setup signals. This subsystem is shown in Figure 3-12 without the control circuitry.

Not mentioned above are the data storage stacks which were seen in Figure 3-9. These will be provided off chip and thus will not be described here.

The chip set will operate as follows. The reader may wish to view Figure 3-13 as an overall picture of the Quantized Sinusoid DFT hardware to help visualize the data movements and control signals. An initial reset cycle will be described, followed by a complete clock cycle.

Upon synchronization of the reset signal and the inactive phase of the clock,  $\phi_2$ , the output of the phase accumulator is zeroed, and a new phase increment is loaded. The zeroed phase accumulator output is latched into the decoder latches. During this same cycle, the coefficient accumulator is cleared and the output latches receive the

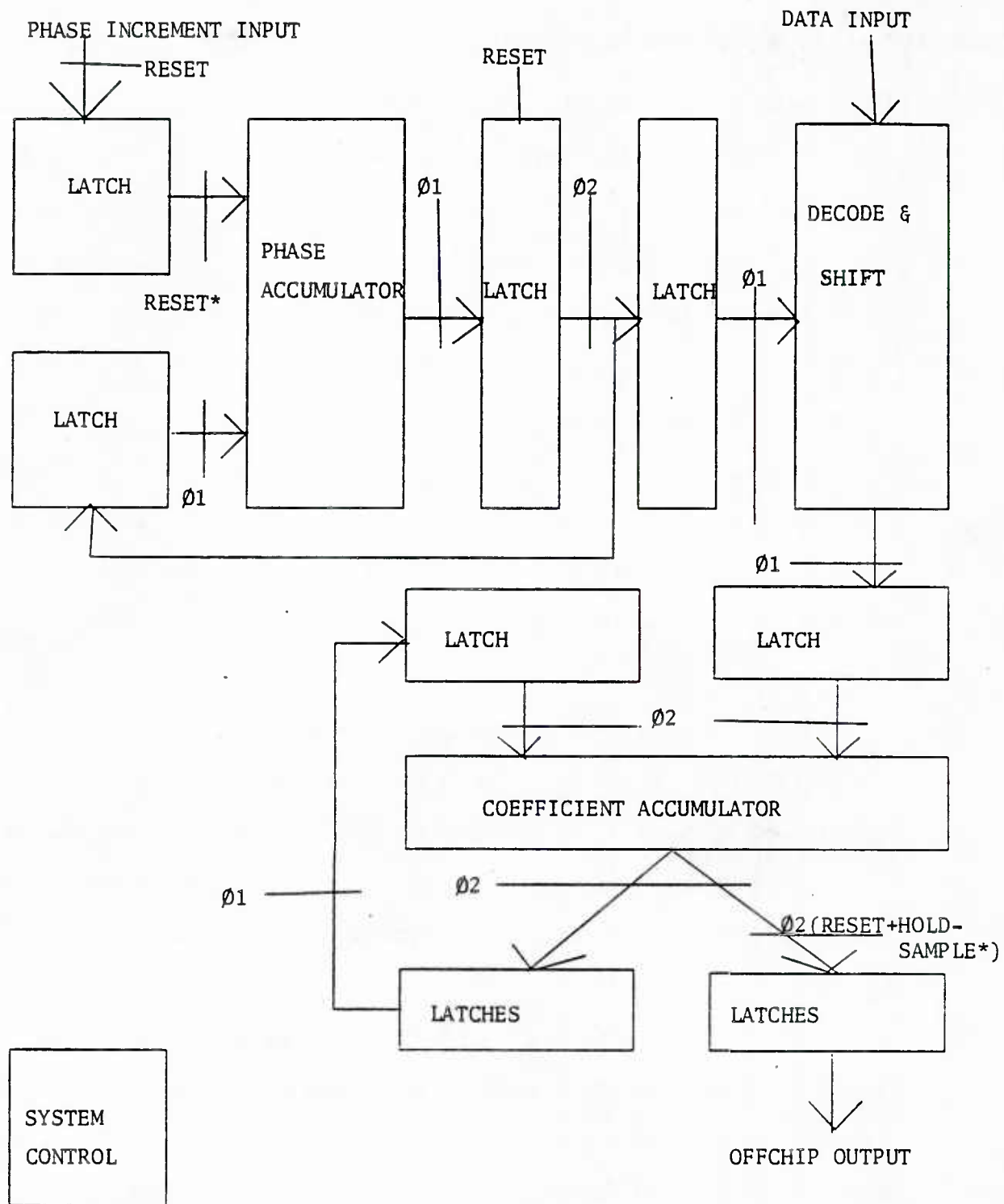


Figure 3-13  
Block Diagram of Total Quantized Sinusoid System

newly calculated coefficient. This cycle sets up the system to begin calculating the next coefficient.

On the active portion of the system clock, **phil**, the phase increment will be accumulated and latched in, ready to be sent to the decode circuit. Also, the previously calculated phase is decoded and is used to set the multiplexor network and the 2's complementer which is used for subtraction if needed. The next data value is shifted through the multiplexor and 2's complementer and is latched into the input buffers of the coefficient accumulator. The previously calculated partial coefficient is refreshed in the coefficient accumulator output latches.

**Phi2** without a reset allows the coefficient accumulator to update its output value. Also, the system output may change if allowed by the **hold\_sample\*** signal. While the coefficient accumulator is updating, the phase value which was calculated during **phil** is latched into the decode buffers. This phase is also refreshed in the phase accumulator. The cycle repeats until a reset is generated by the control circuitry, or by an externally applied reset command.

From the above discussion, it is expected that the adder units will be the factor which determine the overall

operational speed of the system. The decode time during **phil** will be small when compared to the time required to calculate the next phase value, thus the time required to input and "multiply" the data has been made invisible to the rest of the systems operation time. Preliminary investigations show the operation time for a single bit of the adder to be used as 10 nanoseconds, thus requiring 100 nanoseconds for **phil**. Similarly, **phi2** will have to be 200 nanoseconds. This gives an overall operational clock limit of 3.33 Megahertz.

The chipset described above also has testing circuitry built in. The outputs of the phase accumulator may be configured on chip as a serial shift register. This will enable the direct inspection of the phase accumulator outputs. This shifter may also be used to quickly insert test data to the rest of the system beyond the phase outputs. A similar shifter will be inserted between the multiplex network and the coefficient accumulator. This will allow for inspection and insertion of test data in this section of the system as well. In addition to the serial test sites, several inspection ports have been designed into the chips. These will allow for the viewing and setting of internal nodes.

## Summary

A method of quantizing sinusoids which provides effective digital filtering is given. An efficient way to decode the quantized sinusoids was determined. Useful data representations were determined. The operation of the algorithm was discussed. A description of the hardware and its operation and testability was given. Operational limits of the system were discussed.

## CHAPTER 4

### VLSI SPECIFICATION OF THE QSDFT

The QSDFT which was expressed in Chapter 3 must now be implemented in a VLSI technology. The VLSI design will be presented as follows. After a brief overview of nMOS design techniques, the circuit diagrams and nMOS layouts of the basic cells will be described. These will then be consolidated into a three chip integrated circuit set. The circuit schematic diagrams and layouts will be presented at the end of the chapter.

#### Basics of nMOS Design

As with all VLSI mask level design methods, the nMOS design process uses geometric shapes to distinguish the fabrication steps to be taken upon the silicon wafer. In the simplest designs, the geometric primitives are rectangles or boxes. Each box is assigned a color to represent the associated fabrication step. A list of the nMOS fabrication processes and the corresponding color designations is given in Table 4-1.

Given the boxes which represent the fabrication steps, a designer may then specify the geometric patterns which will produce the desired circuit. The various electrical components of the circuit will be created when the corresponding geometric areas are transferred to the silicon wafer. For example, a transistor is generated when red (polysilicon) and green (diffusion) boxes intersect. A resistor can be a length of any of the conductor layers (polysilicon, diffusion, metal).

---

<u>Layer Name</u>	<u>Layout Color</u>	<u>Use</u>
glass cut	orange	creates an opening for bonding pads
ion implant	yellow	reduces resistance of diffusion
		changes conductance of transistors
buried contact	brown	connects poly and diffusion
contact cut	black	connects metal to poly or diffusion
diffusion	green	conductor layer
polysilicon	red	conductor layer
metal	blue	conductor layer

**TABLE 4-1**  
nMOS Fabrication Layers

---

The specification of the geometric patterns may be accomplished via an interactive layout editor or created manually. The fabrication specification used is the Caltech Interchange Format - CIF [15]. The geometric layouts and CIF

files for this project were generated using the Circuit Hierarchy Integration Program (CHIP) [16], an interactive layout tool developed at the University of Rhode Island. Once the CIF definition for the project has been established, the CIF file is sent to the MOS Implementation Service (MOSIS), in Marina Del Rey California for fabrication. The final IC's are then returned to the designer for testing.

#### Basic Cells for the QSDFT

The basic cells which were used in assembly of the QSDFT chips will be listed as follows. Those cells which were derived from standard designs will be listed first. Those cells which were designed completely by the author will then be covered in more detail.

The primary cell in the QSDFT chips is the adder. It is used in both the phase detection as well as in the generation of the filter's output. A simple (and correspondingly slow) design was chosen for the adder in order to simplify the task of debugging the entire system. Its design follows from the ripple carry adder described by McCormick [17]. A simple programmable logic array is used to decode the three inputs  $A_i$ ,  $B_i$  and  $Carry\_in_i$  and their inverses. The outputs are  $Sum_i$ , and  $Carry\_out_i$ . Its nMOS

transistor circuit equivalent is shown in Figure 4-1. An nMOS layout is given in Figure 4-2.

The next basic cells to be described are the buffers. Several varieties were developed from the basic layout given in the VLSI Designer's Library [18]. These buffers employ a series of inverters which increase in size to give the desired output drive capability. Figure 4-3 shows the circuit diagrams for the extra-large sized buffers. The layouts are shown in Figure 4-4. Buffers of other sizes may be created from the extra large buffers by changing the transistor sizes.

Also used in the QSDFT chips is a PLA. The PLA cells were also derived from [18]. In addition, a FORTRAN77 routine was written to generate a pre-programmed PLA from the basic cells. The programming is accomplished by reading in the desired truth table, such as the pattern given in Table 3-1 for the sinusoidal decoding. The layout of the sinusoidal decode PLA as generated by the MAKEPLA routine is shown in Figure 4-5.

Another basic cell which was used in the QSDFT chips is a cell which generates a non overlapping clock from a single phase clock. This cell is described by Mead and Conway [15]. It consists of an inverter and two nor gates. Its circuit

diagram and layout are given in Figures 4-6 and 4-7 respectively.

The following cells were designed from the ground up for use in the QSDFT project. They all entail the ideas of the QSDFT in that they seek to perform a specific function in a minimum of silicon area. This leads to extra design time, but inherently smaller and more compact basic cells.

The first custom cell to be described is the data multiplexor circuit. It consists of a pass transistor network to steer the data through correctly. Sign extension and the shifting in of the least significant bits is also accomplished in the cell. The MSB of the input data is anded with a signal which denotes two's complement data. If the two's complement signal is high, the MSB of the data is sent though to the multiplexor as is needed with two's complement data. The original MSB is sent to be exclusive or'ed with the MSB output of the phase adder. These two MSB's determine if the product of the quantized sinusoid and the data is positive or negative. The output of the the multiplexor is then inverted or left unchanged depending upon the sign determination. A negative product causes the shifted data to be inverted to cause an effective subtraction in the coefficient accumulator. The sign signal is also used as the carry in to the coefficient accumulator as the final

operation in the subtraction process. This layout is detailed in Figure 4-8.

The next cell to be described is a basic latch. This cell was designed to be used in a first in - first out stack, as well as a general purpose static data latch. The cell may also be concatenated to form a master slave data latch. The design includes two inverters and two pass transistors - one of which acts as a feedback path, the other controls the data path. A diagram of a master slave latch made from these latches is shown in Figure 4-9.

The master slave latch operates as follows. When  $\phi_1$  is low ( $\phi_2$  high) the new input data traverses the two inverters. Simultaneously, the previous value is presented to the outputs and is recirculated through the feedback pass transistor. As the clocks switch states, the new data is latched into the output half and is recirculated in the input stage. A new  $\phi_2$  high signal permits the newly latched data to be presented at the outputs.

The need for the recirculation transistor can be seen from the circuit layout in Figure 4-10. If one of the stages had been set to a logical zero level when a logic one is written in, the charge representing the new signal would be sunk to a ground potential through the pull down transistor

of the second inverter. The recirculation pass transistor prevents this path from occurring during the latching process.

The last basic cell to be described is that of the counter. The counter uses the basic concept of binary counting. That is, any given bit in a binary counter toggles if and only if all bits of lesser value have been set. This means that the output at the  $i^{\text{th}}$  bit will be the exclusive-or of the bit's previous value and of the product of all the less significant bits. This design allows for a much smaller cell than if flip flops had been used. A limitation of this cell is that the count may not reverse direction during the count cycle. ( The outputs may be inverted to give a down counter.) Also, a preset capability is not given for this cell. A reset is permitted though, making this counter useful for control circuitry.

The operation of the counter may be described using Figure 4-11. After a reset command, the first clock pulse enters the system. The  $\phi_1$  represents the input clock signal and  $\phi_2$  is its inverse as generated by the two phase clock cell.  $\phi_1$  allows the outputs to propagate through the inverter latches. Since  $\phi_2$  is low, the new outputs may not affect the next stage at this time. Once  $\phi_1$  becomes low and  $\phi_2$  is high, the new value of the bit is passed onto

the next stage. The new value of the bit is also presented to the exclusive-or gate in the same bit to begin generating the next value of the bit. This continues until a reset command is issued. Notice that if phil is left asserted for a long period of time, the charge on the AND gate inputs will decay and the count may become erroneous. If, however, phi2 remains high for a long period, the previous data will remain intact in the register and the counter's data will stay valid. This register is one half of the previously discussed fifo basic cell. A layout of a two bit counter is shown in Figure 4-12.

### Chip Set Construction

The three QSDFT chips may now be constructed from the previously described basic cells. The three chips have been shown as functional block diagrams in Chapter 3. The nMOS layouts and the plans for testing which have been built into the chips will be discussed. The multiplexor unit will be discussed first, followed by the phase accumulator, and the coefficient accumulator and control circuitry.

The heart of the multiplexor chip is the pass transistor shift network. The input lines are inverted before being multiplexed to assure signal quality through the pass transistors and into the two's complementer. Since the

outputs of the shifter are inverted, the data must be ex-nor'ed with the sign signal to provide the correct input to the coefficient adder.

The decoding PLA and phase latches comprise the remaining circuitry. The PLA was shown previously. The latches are two stage master slave latches created from a pair of the basic latch cells. The buffers are extra large sized buffers. The two clock phases are generated via the two phase generator and buffered by a pair of large buffers. This same unit was used whenever the two phase signals were needed. This allows for only the phase phil to be broadcast to the chips thus reducing the chance of clocking problems.

Several test facilities were built into this chip. The primary concern is that the PLA may not have been fabricated correctly. To this end, additional pins were allocated to give access to internal points within the chip. The outputs of the PLA may be observed, and may be set externally so that if the PLA itself does not function properly, the shift network may be tested individually.

A final plot of the pass transistor shift network is shown in Figure 4-13. It occupies a 40 pin package. The excess unused area was due to a restriction in the size of forty pin packages allowed by MOSIS. The area of the project

is about 1000 by 700 microns. The overall chip is 4600 by 3600 microns. The project on this chip contains approximately 300 transistors.

The second chip in the chip set is the phase accumulator system. It uses a 10 bit adder to accumulate the phase of the reference waveform. Its inputs come from two registers. The current phase increment is input through one register, the accumulated value is in the other. The output is stored in a separate storage register. The two phase clocks are generated with the same two phase generator as was used in the multiplexor chip.

The testing of this chip is facilitated by the inclusion of a serial shift register which doubles as the output storage register. Since the outputs of this subsystem will be totally internal to a final chip which houses all three subsystems, a method of looking at these outputs was required. The shiftable output register also allows for the insertion of values which will be input to the subsequent system components. When this register is not being used as a test port, it will function uninhibited as the output register of the phase accumulator.

In addition to the shiftable output register, several other test points were include in this chip. These accesses

will permit the monitoring of vital internal nodes in the circuit's structure. As with the multiplexor chip, these nodes may be either observed, or set, depending upon the test requirements.

A final plot of the phase accumulator chip is shown in Figure 4-14. It occupies a 64 pin package. The area of the project is about 2800 by 300 microns. The overall chip is 6900 by 6800 microns. The project on this chip contains approximately 600 transistors.

The final chip in the set contains the coefficient accumulator. A 20 bit adder is used as the basis for the accumulator. The inputs are stored in parallel latches. The output is stored in a dual output register configuration. One of the output latches sends its data back to the accumulator inputs. The other register is used to store updated coefficients. Thus, the outputs may be viewed by the user as often as desired. As with the other chips, the two phases of the control clock are generated via the two phase clock generator circuit.

Testing of the coefficient accumulator will be conducted similarly to the phase accumulator. Internal test points will be observed. In addition, a shift register may be used to shift in new inputs to the coefficient accumulator. In

the non test mode, this register will pass information from the multiplexor directly to the adder's input latches. This test mechanism was not built into this chip, but will be implemented when the entire system is fabricated on a single chip.

A final plot of the coefficient accumulator chip is shown in Figure 4-15. It occupies a 64 pin package. The area of the project is about 5600 by 300 microns. The overall chip is 6900 by 6800 microns. The project on this chip contains approximately 1400 transistors.

### Summary

A basic review of nMOS design was given. (See Introduction to VLSI Systems [15] for more information.) The basic cells of the QSDFT were described. These cells were then assembled into a three chip set to implement the Quantized Sinusoid DFT algorithm which was specified in Chapter 3.

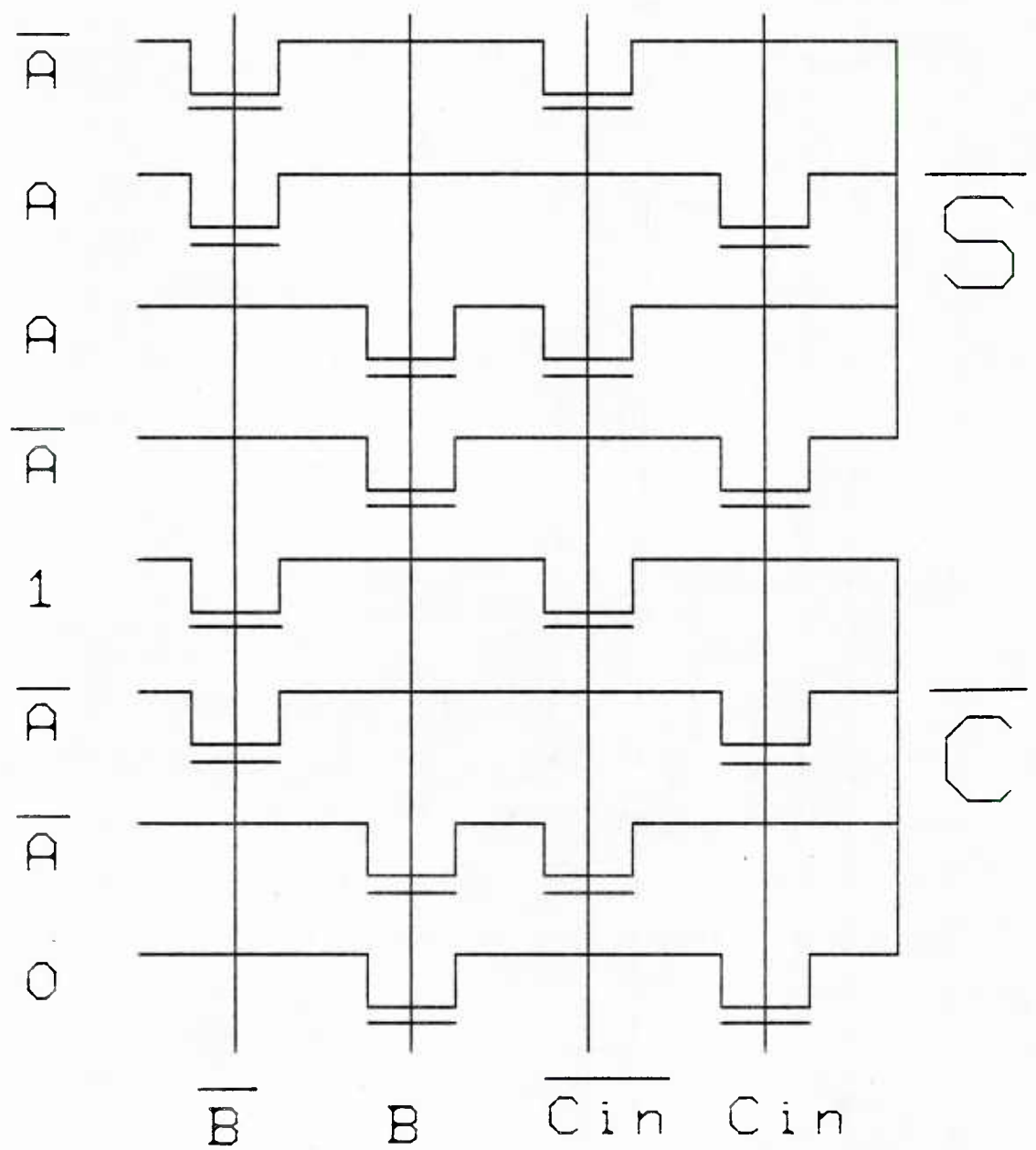


Figure 4-1  
nMOS Adder Transistor Schematic

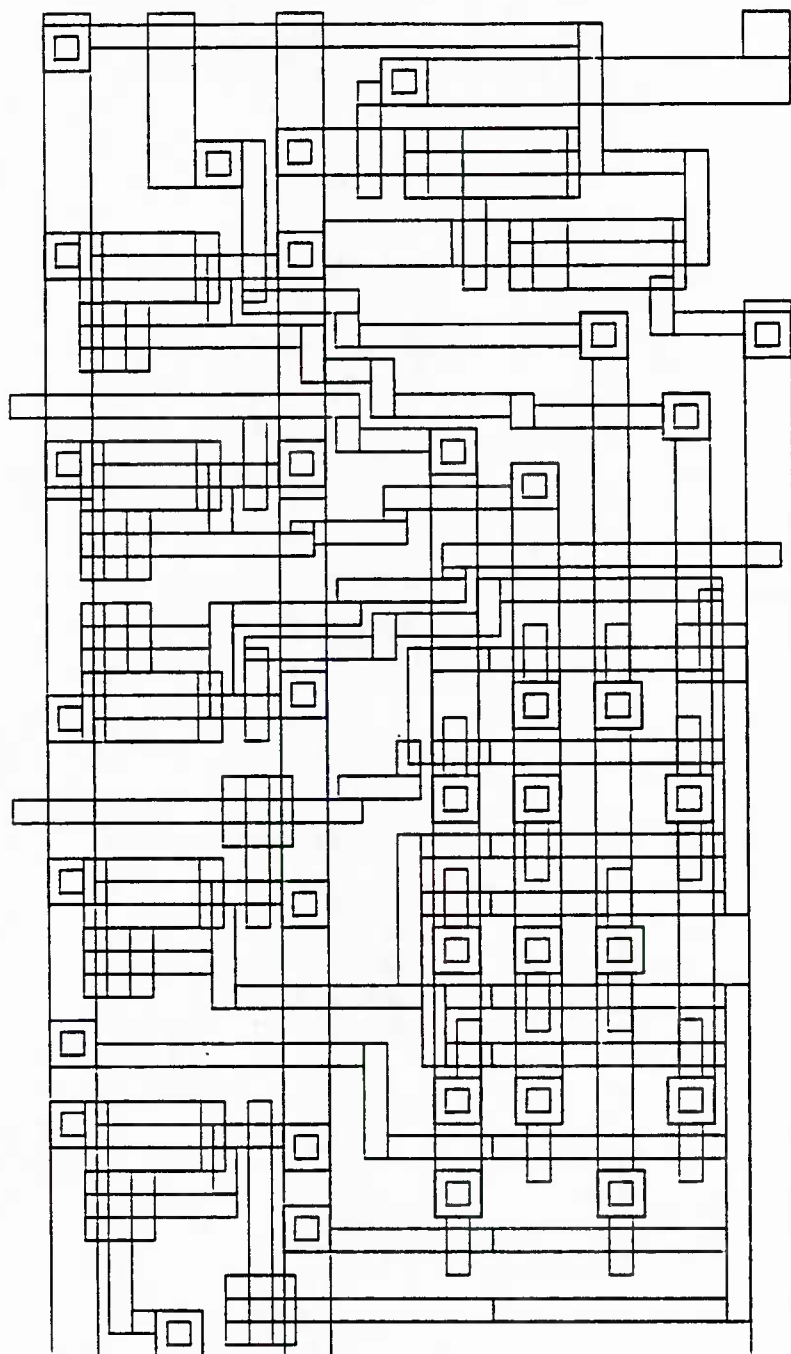
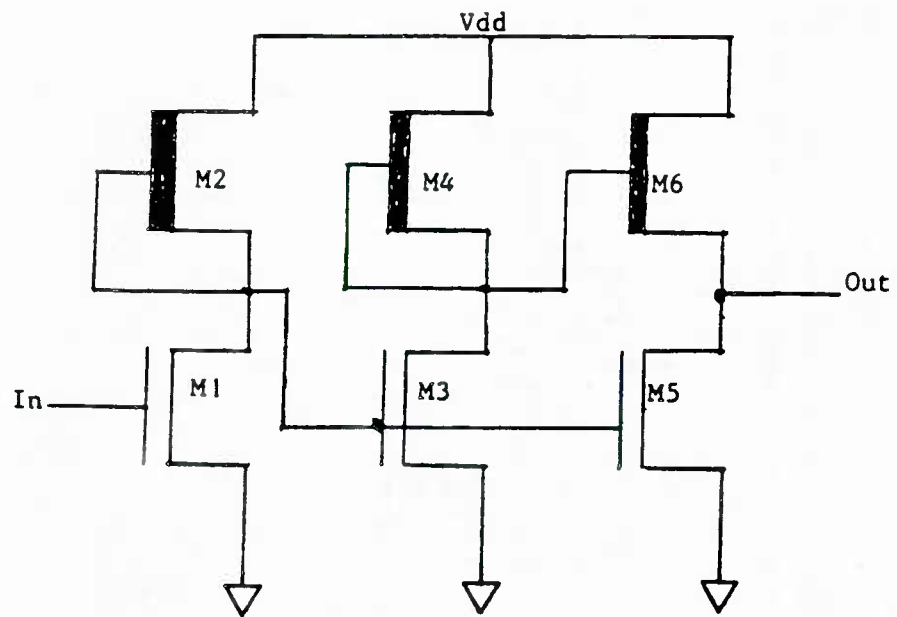


Figure 4-2  
nMOS Adder Layout

NON-INVERTING



INVERTING

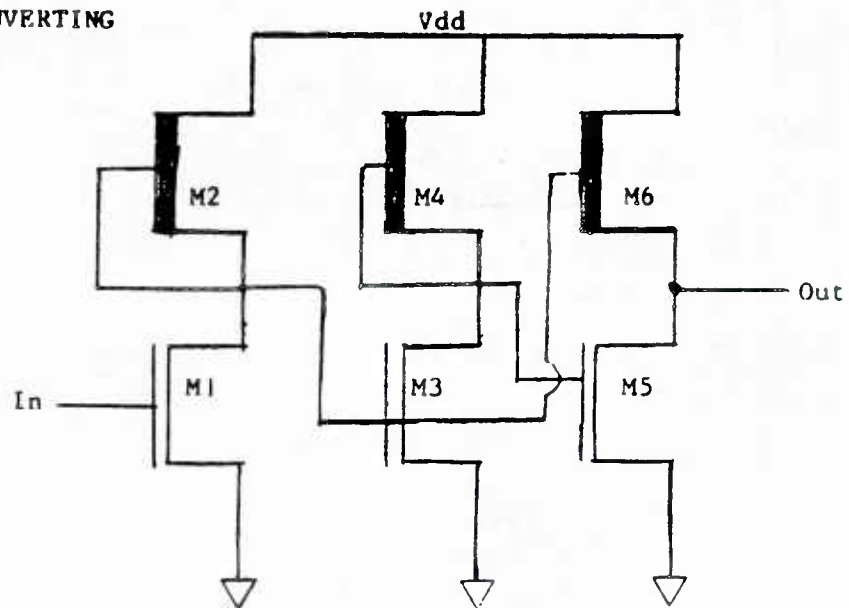


Figure 4-3  
Extra Large Buffer Schematics

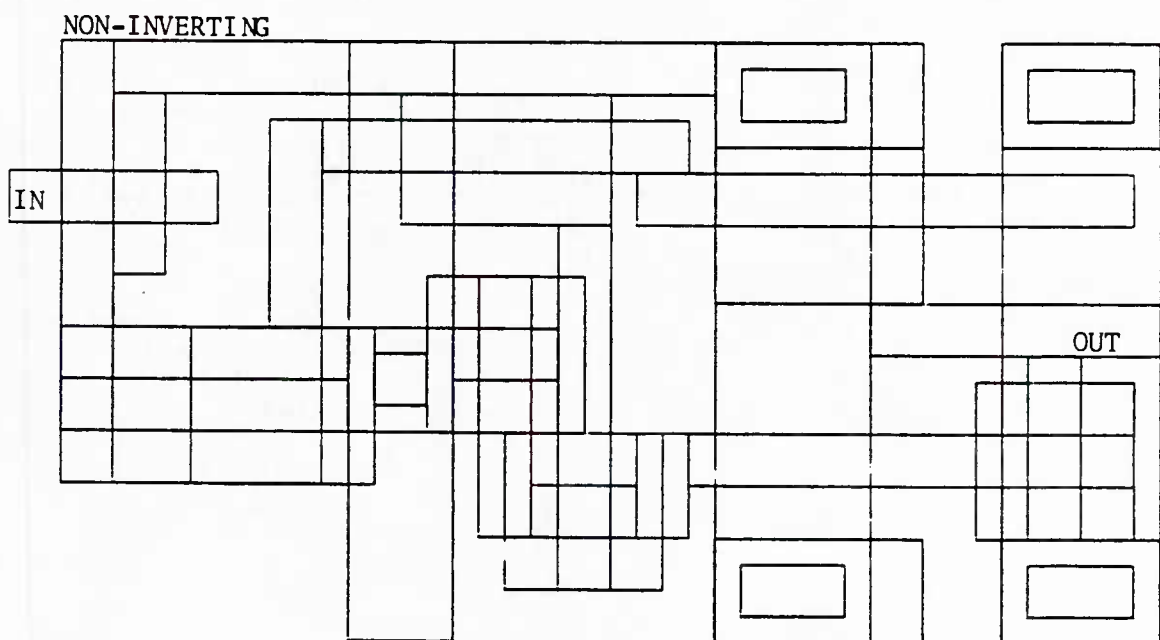
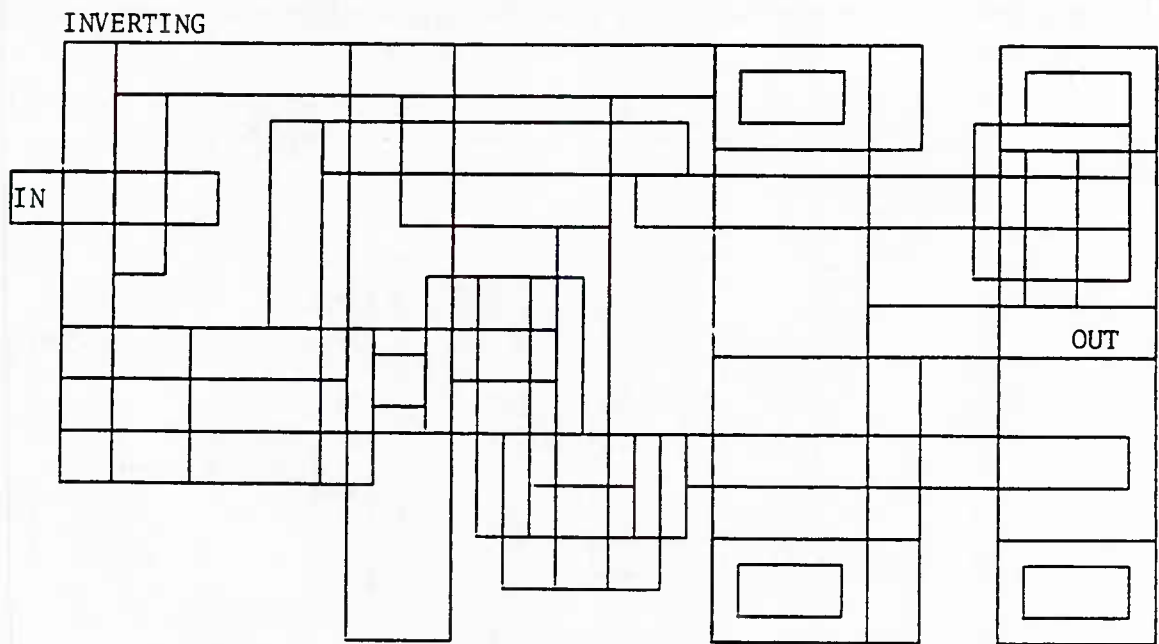


Figure 4-4  
Extra Large Buffer Layouts

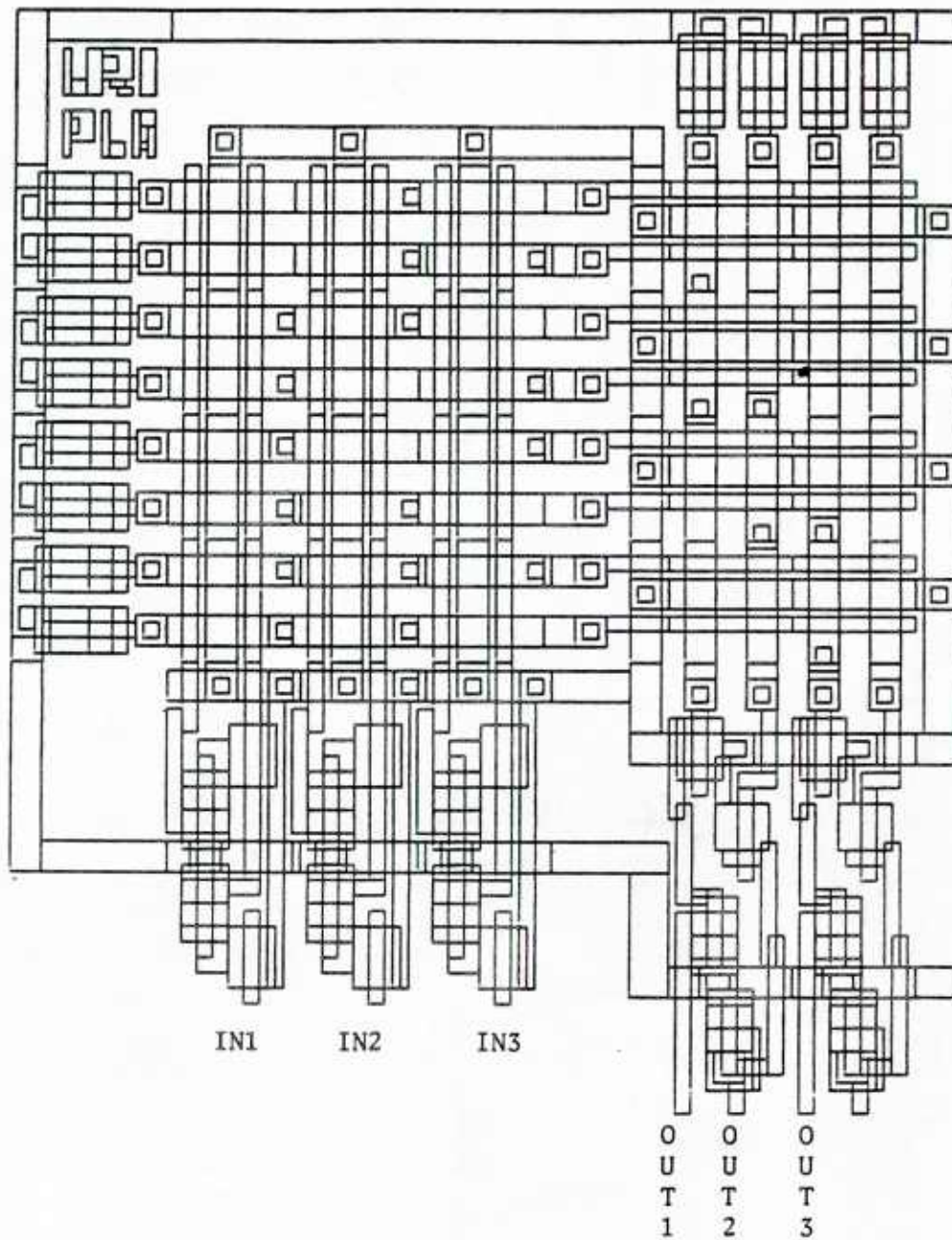


Figure 4-5  
nMOS PLA Layout

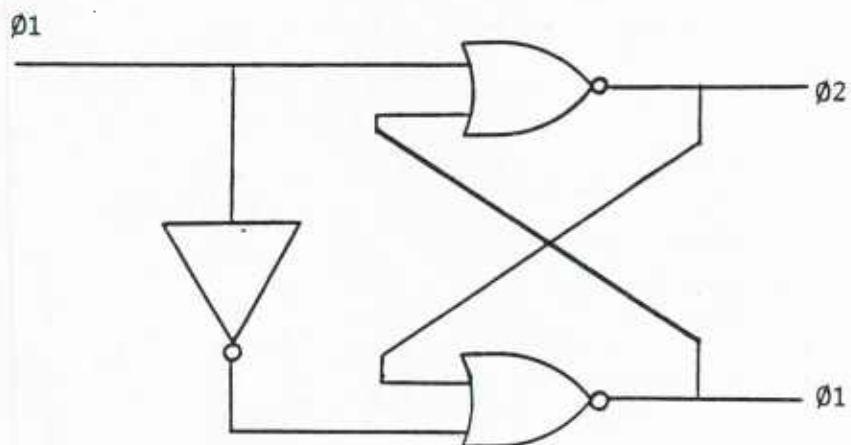


Figure 4-6  
Two Phase Clock Schematic

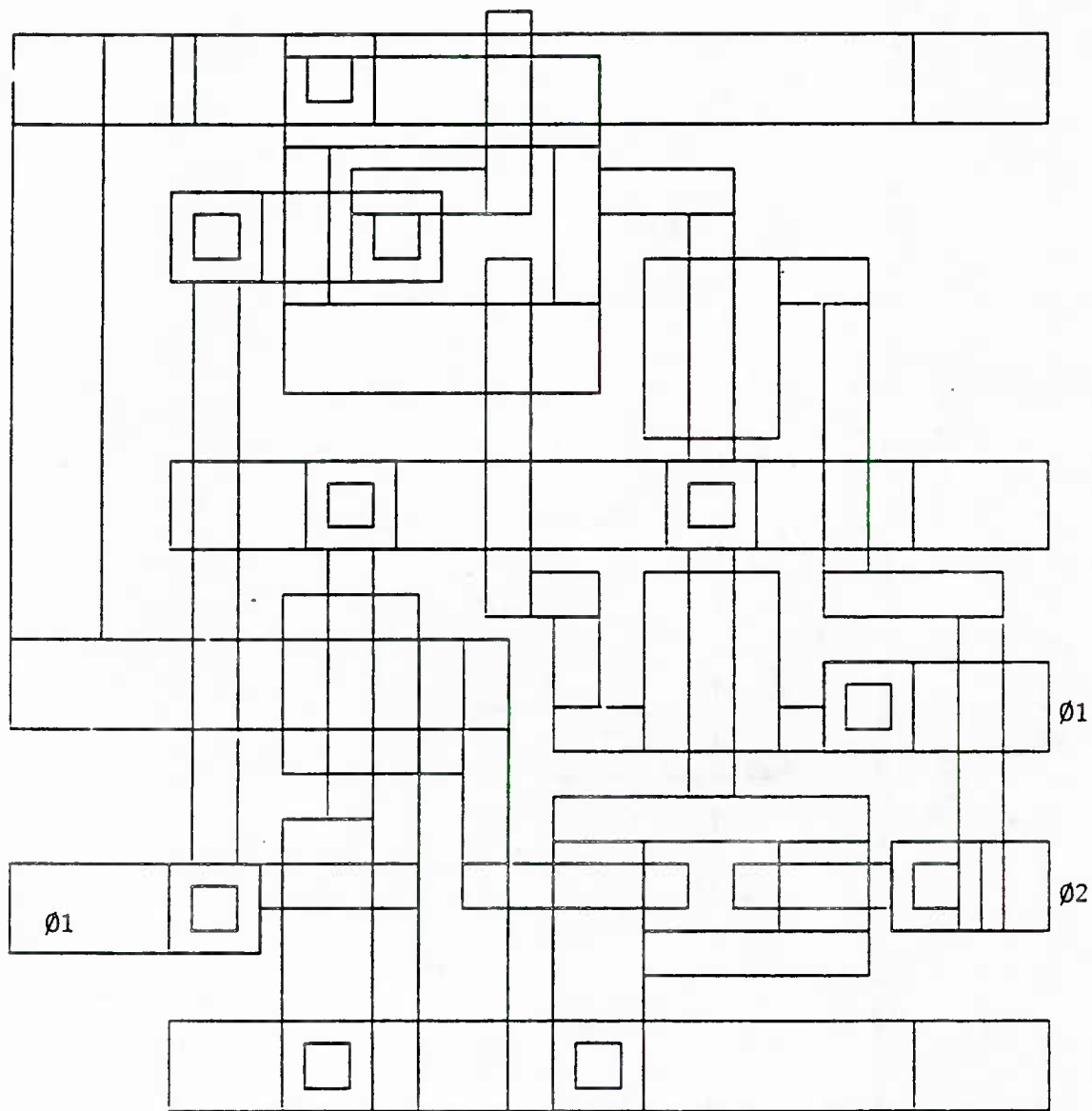


Figure 4-7  
Two Phase Clock Layout

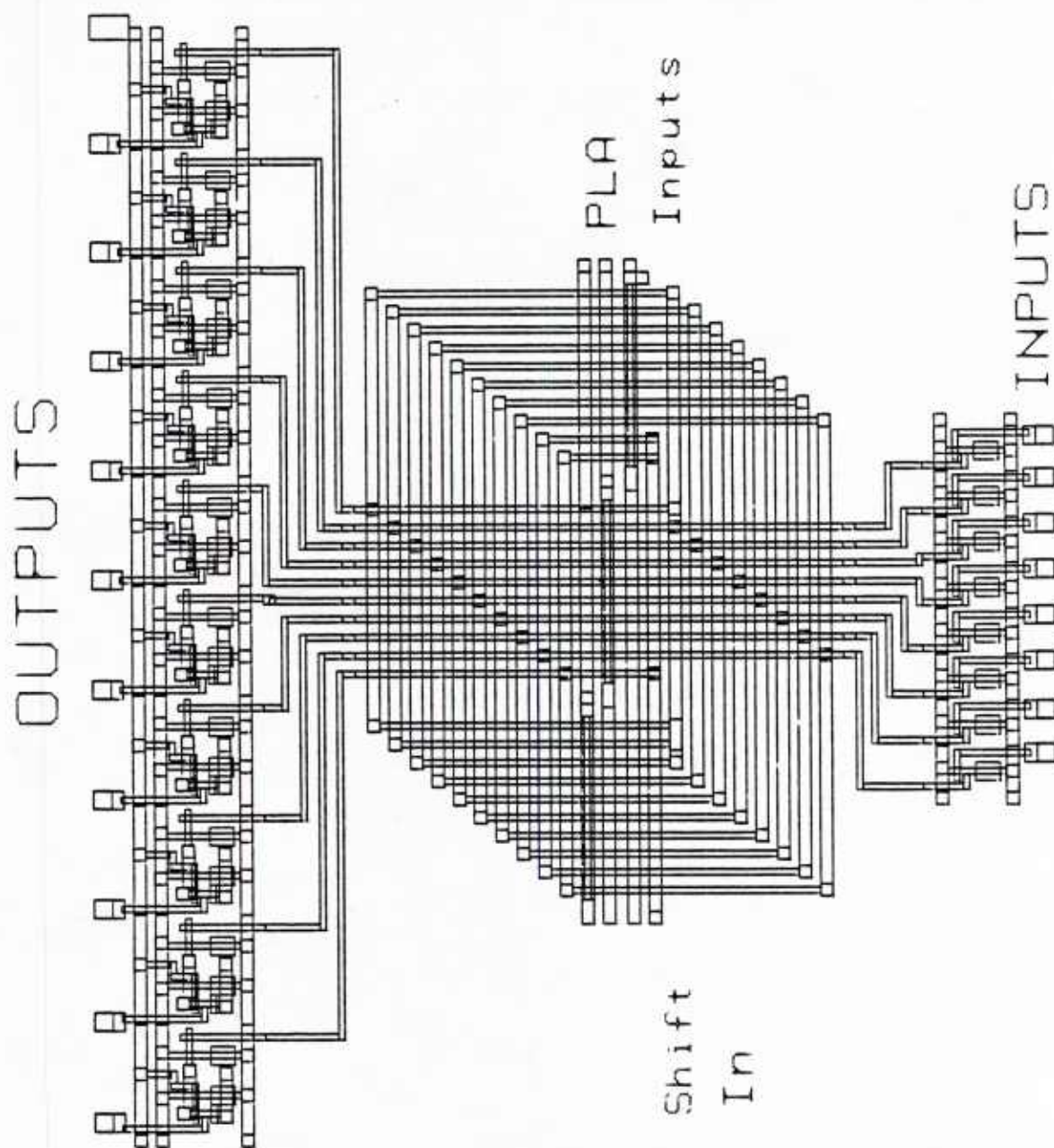


Figure 4-8  
Pass Transistor Shift Network Layout

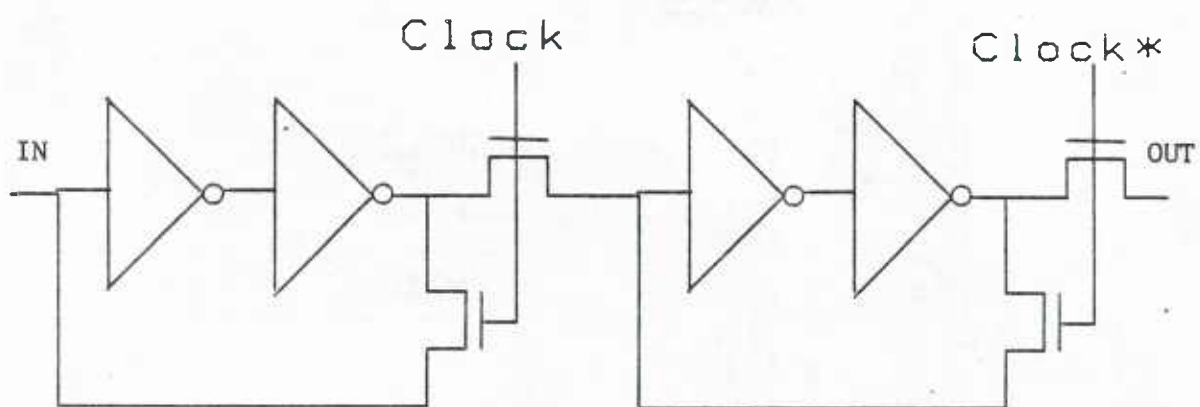


Figure 4-9  
Latch Schematic

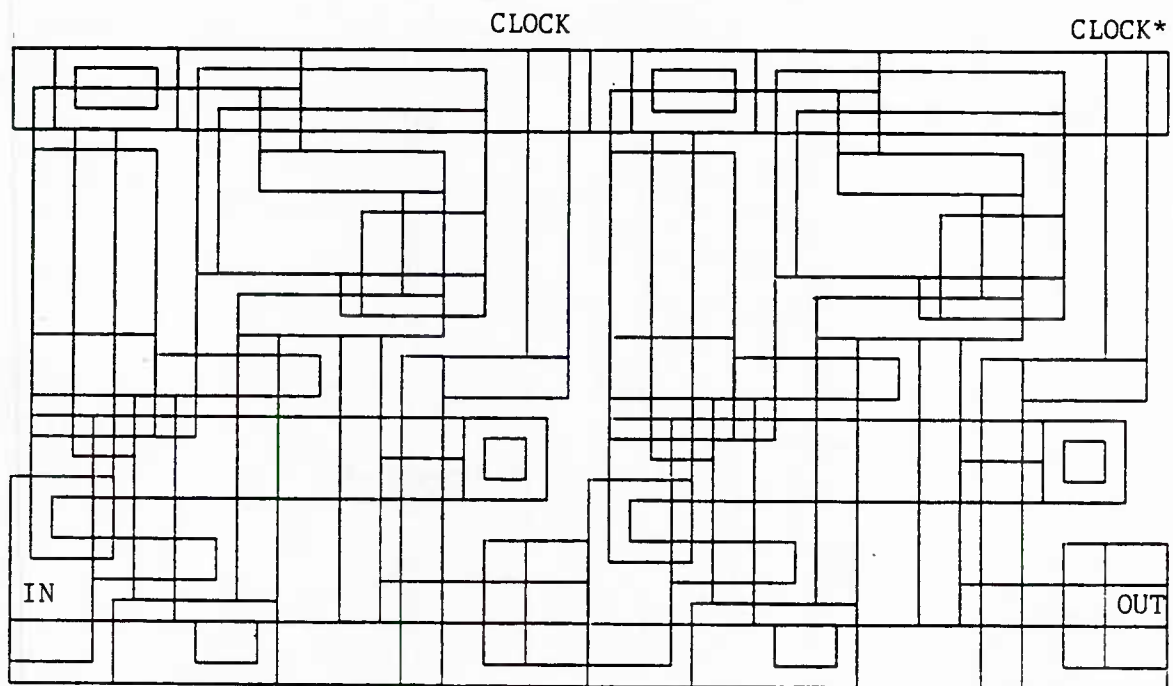


Figure 4-10  
Latch Layout

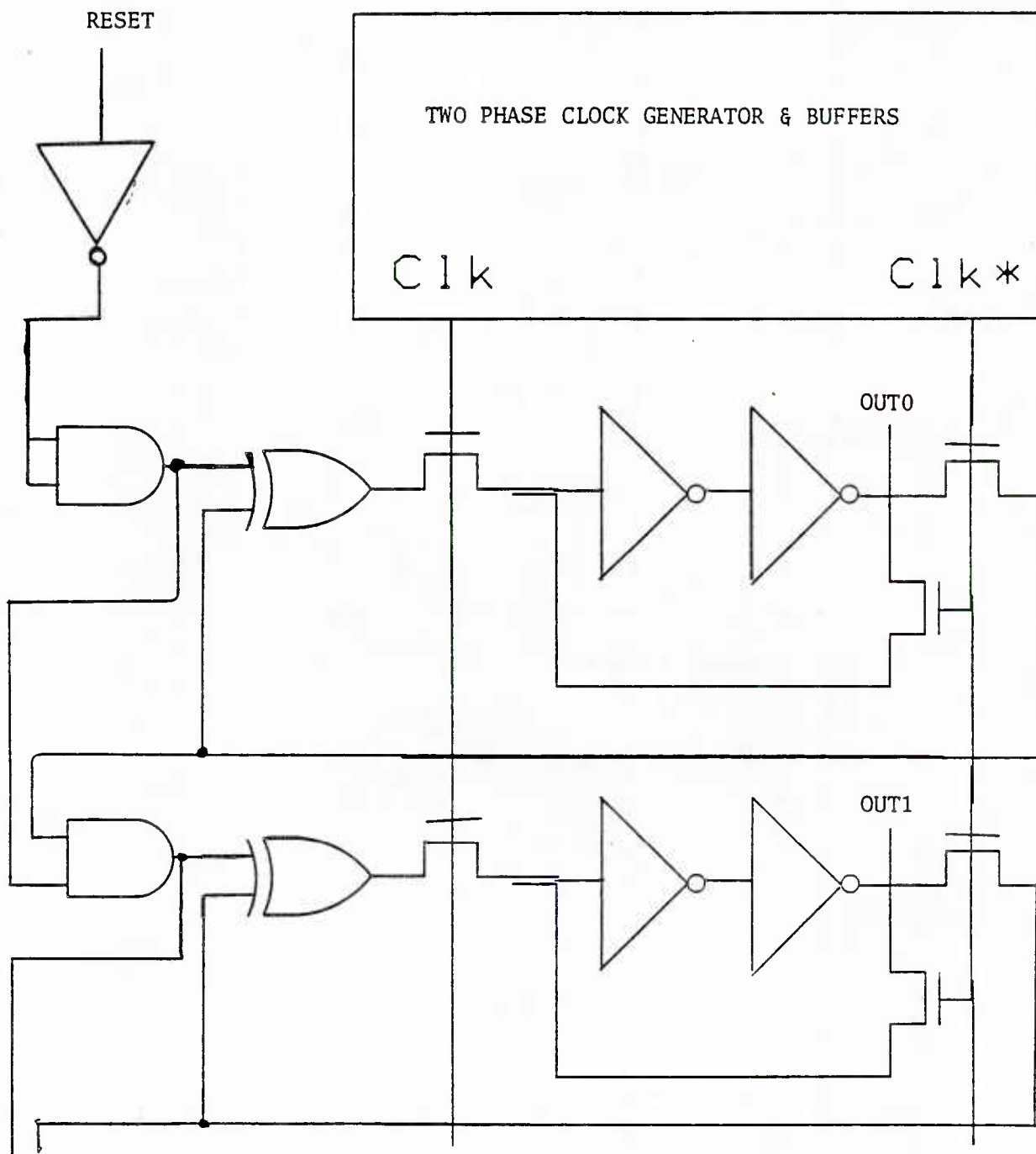


Figure 4-11  
Counter Schematic

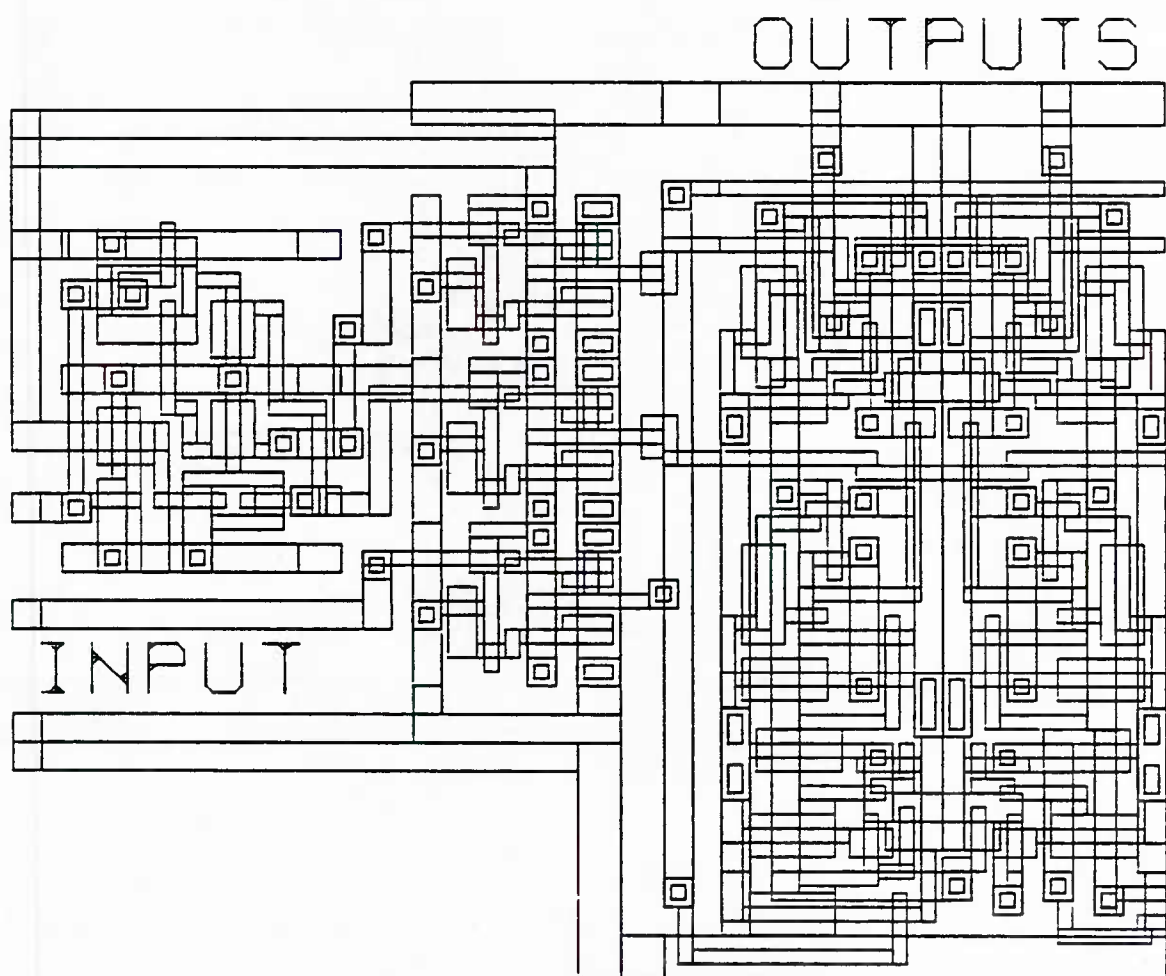


Figure 4-12  
Layout of Two Bit Counter

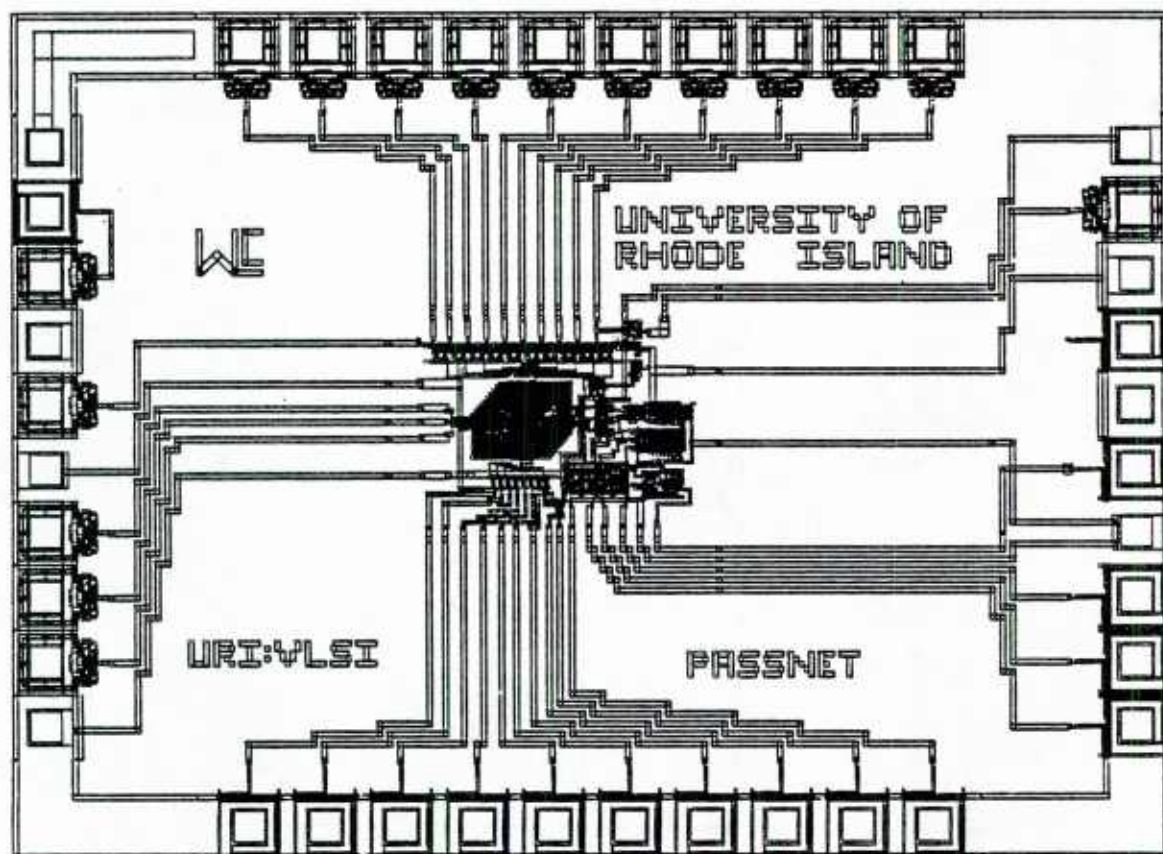


Figure 4-13  
Final Plot of PASSNET

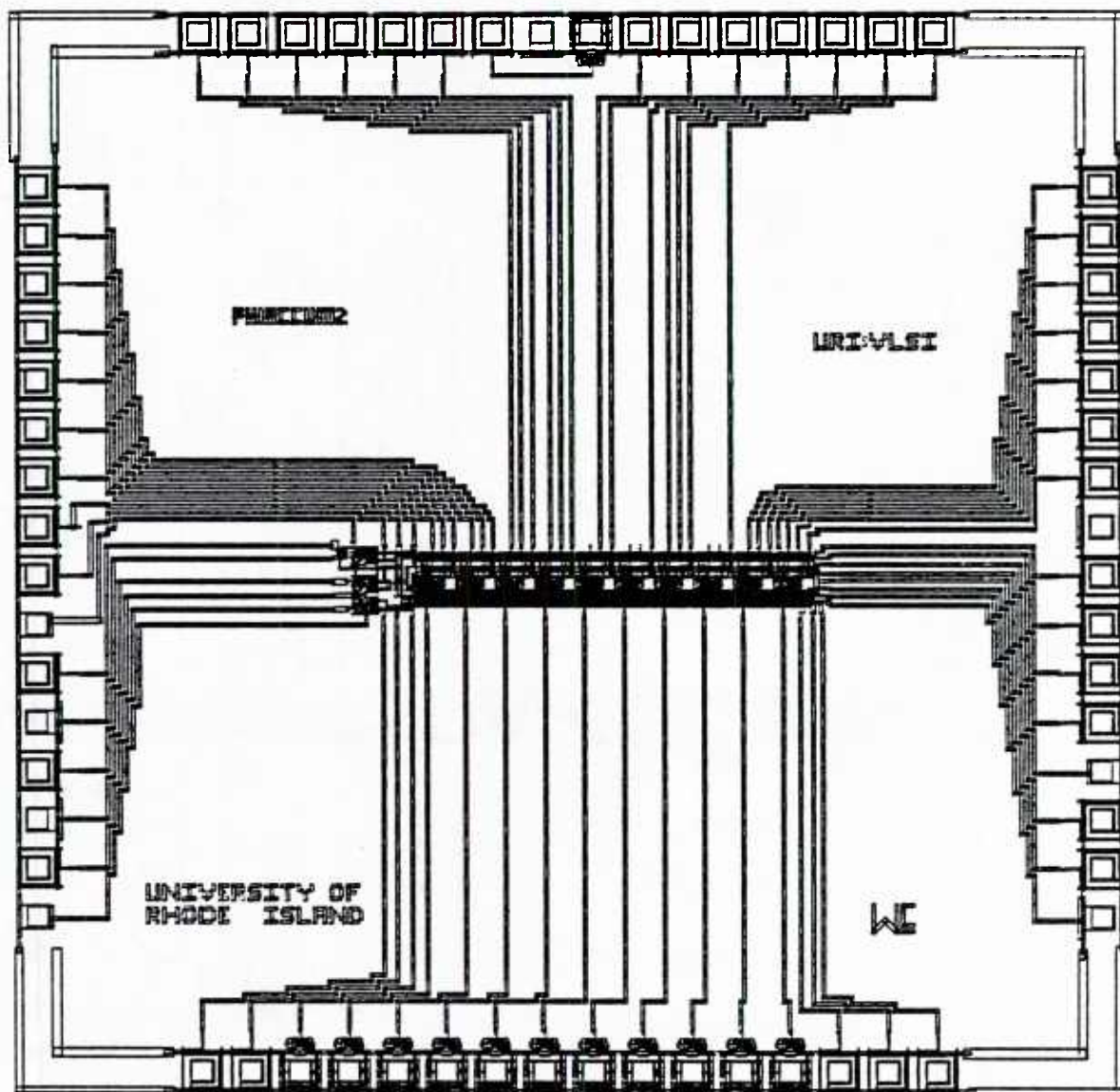


Figure 4-14  
Final Plot of PHIACCUM2

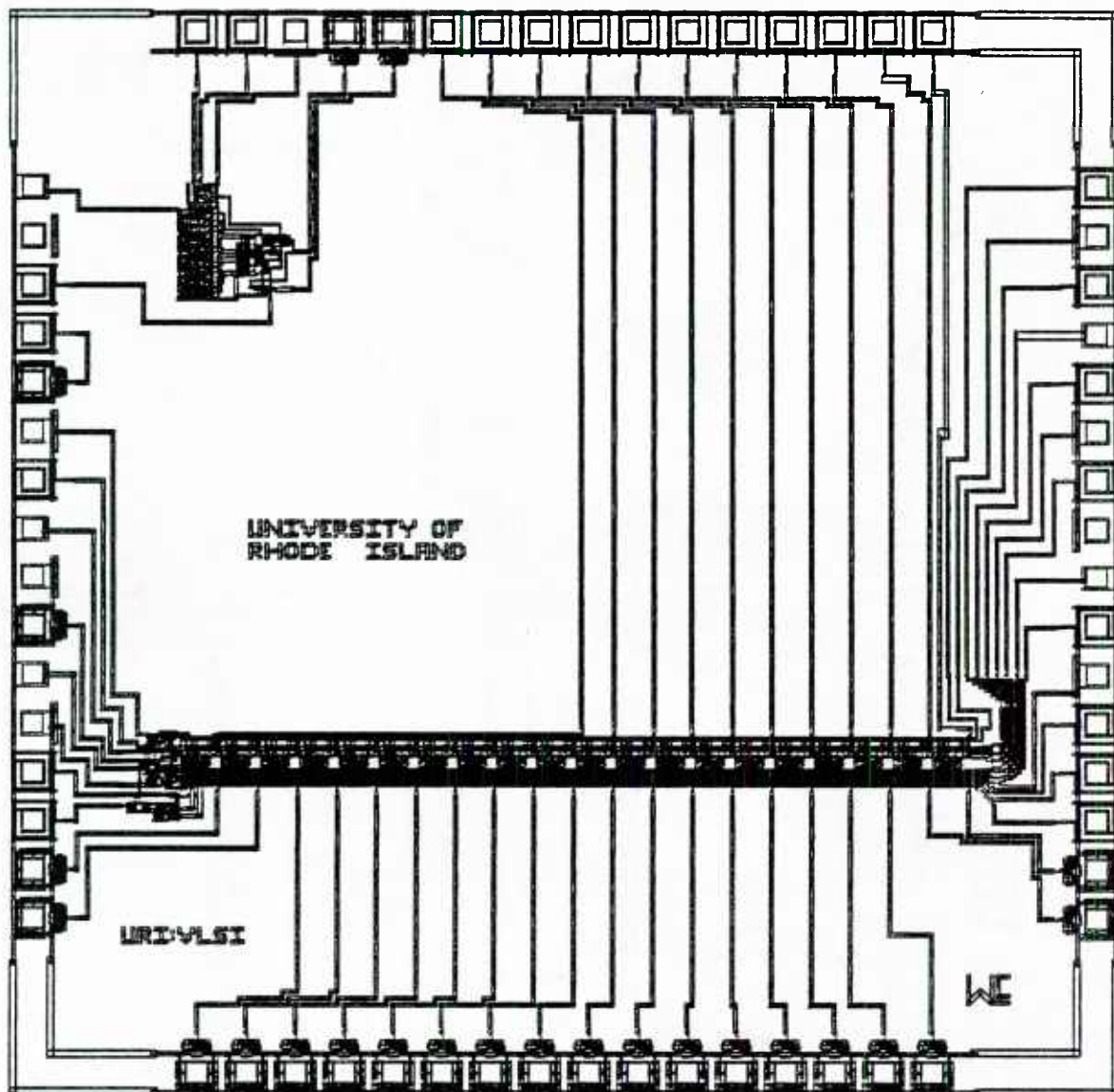


Figure 4-15

Final Plot of COEFFACCUM

## CHAPTER 5

### HARDWARE TESTING AND RESULTS

Once the integrated circuits have been fabricated, they must be tested to determine if they are operational. If the IC's are found to be defective, several stages of testing are done to determine if the error is in the fabrication process or the design and layout stages. After a brief discussion of the testing techniques to be used, the test results of the components used in this project will be given.

Initially, low speed 'DC' tests are made to ascertain if the fundamental parts of the chip are correct. These tests check functionality of the input and output pads, basic cells on the chip, and portions of larger cells clocked at speeds much slower than normal operation. In addition, parameters such as current requirements and switching speed limits may be measured to provide an indication of the chip's overall chances of functioning properly.

Once the chip passes the 'DC' tests, full testing is done. The purpose of this step is to determine if the circuit functions completely as planned. This functional

testing involves exercising the circuit to such an extent that it can be assured that the system is operating as designed.

Full functional testing does not, however, imply that the chip must be exercised to such an extent that it passes through every possible state. For example, it suffices for an adder to see if all bits of the adder are operating correctly and that the carries between bits are also correct. Other in chip test methods may be used as well to 'look' inside the chip. These ideas will not be expounded upon here. Instead, several references on the subject are provided in the bibliography.

Several tools were assembled to aid in the testing of the circuits. General items such as logic analyzers, oscilloscopes, and current and voltage meters were used. A special device which was also used was a test system built at URI which is based upon a Motorola MC68000 VERSAbus computer. The VERSATest machine allowed for transmitting test vectors up to 32 bits long to and from the chip under test at speeds of 100 kHz.

Before any basic cells could be tested accurately, a measurement of the single inverter delay had to be made. This was accomplished with a circuit whose primary function

was to serially convert a parallel number to its two's complement. The circuit consisted of a string of NAND gates and inverters. Since the number to be converted could be held constant, an accurate measurement of the delay through the inverter and NAND gate could be made by observing the clocking of the first bit with the others held steady. This circuit is shown schematically in Figure 5-1. The measurement revealed a basic delay of 4 ns through the inverter when driving a load of two similar gates. As expected the NAND gate, which effectively is twice the size of the inverter, had an 8 ns delay. These measurements can now be used as a basis for determining how well the other basic cells operated by scaling these numbers according to the actual device sizes.

### Testing of the Basic Cells

The adder unit was the first basic cell to be tested. For test purposes, eight cells were concatenated into a string. This allowed for convenient full testing as well as accurate timing measurements. Full functional testing of the  $2^{16}$  possible inputs took about 3.5 seconds. In this case the eight bits could be fully tested almost as fast as if special set of test vectors had been used.

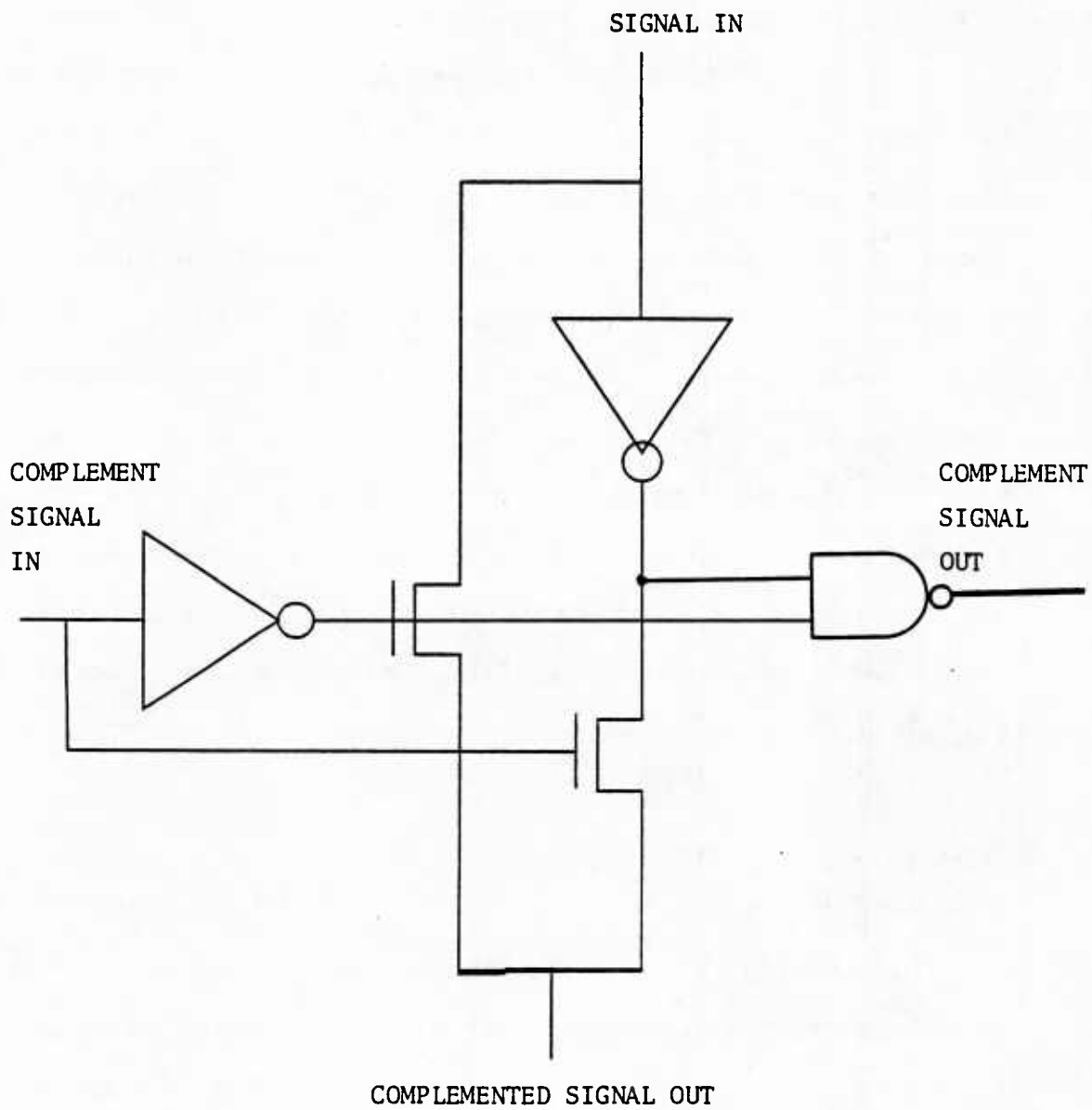


Figure 5-1  
Schematic of Basic Delay Circuit

Additional measurements were made using the string of adders to determine the operational speed of each adder bit. By holding one 8 bit input word high, the other low, and toggling the Carry\_In signal, a delay over the entire 8 bits could be measured. The Carry\_In signal was fed back out of the chip immediately to provide a basis for measuring the delay at each of the bits. Since this signal traversed the output pads, as did the adder output bits, the delay through the output pads was cancelled. An average delay per stage of 10 to 12 nanoseconds (ns) was measured.

The second cells to be tested were the superbuffers. Since these had been converted from existing designs, no timing measurements were made other to see that they would run as fast as the input/output pads. Also, it was ascertained that the designs had been correctly converted from an older nMOS technology to the one presently supported by MOSIS. From the above inverter delay measurement and the known size of the extra large buffer inverters, it can be calculated that the total delay should be approximately 28 ns ( $8 + 4 + 16\text{ns}$  per stage) for a 16 inverter load. This compares well to SPICE (Simulation Program with IC Emphasis) simulations which show a 15 - 20 ns delay when only a simple device model is used.

The next testing to be described is that of the Programmable Logic Array cells. As with the buffers, these cells were obtained from the VLSI Designer's Library. Because of the regular, but expandable nature of PLA's, no explicit timing information can be easily gleaned from them. Again, no speed measurements were made other than a check against the pad limit. Correct operation of the PLA was also observed.

The last 'standard' cell to be tested was the two phase clock generator. This cell uses logic gates to create two non-overlapping clock signals from a single clock. The two phases are then buffered with super buffers to give added drive capability. The basic unit (without buffers or a load) was able to switch as fast as the pads. Separation time between the two signals under a 16 gate load was measured as 70 to 80 ns with a clock speed of 0.5 MHz. The upper limit of the clock with a load and test point loading (extra input pads on the bus) was 2.91 MHz (344ns). At this speed, the separation increased to 115 ns. A much smaller separation is to be expected when the above inverter delay numbers are considered. The reason for this increase in separation was attributed to loading of the circuit caused by the extra test pads which were included in the design.

The first custom cell to be tested was the primitive cell to be used in the first in first out (FIFO) stack. This base cell was tested in two ways. In the first configuration, a single line of the cells was formed as a one bit FIFO stack. The first bit out was used to trigger a logic analyzer with time slices which were equal to the driving clock. In general, each stage of the fifo took less than the input clock period to complete its shift so that each bit toggled in a new analyzer time slice. The fifo, as with the other circuits, operated to the limit of the input pads. In a second configuration, (which was fabricated on several different chips), two fifo cells were used as a master slave latch. This cell also operated to the limit of the pads. Estimated speed is about 20 MHz (or 50 ns per master slave stage) including the clocking signal generation.

The last basic cell tested was the custom designed counter. The cell functioned properly at clock speeds up to 1.5 MHz. However, some bouncing of the outputs was observed at the input clock transitions. These spikes, though, should not affect the overall functionality of the counter at this operation speed.

Another fault detected in the counter's operation was that the edge triggered action usually associated with a

synchronous counter was not evident above the 1.5 MHz range. The first output bit emulated a frequency divider with a positive duty cycle of about 0.37, instead of 0.5 triggered on the input edges. At 3.88 MHz, the active time of the input clock plus the known separation of phil and phi2 from the two phase generator is roughly 200 out of 257 ns. From this, it appears that the feedback signal (see Figure 4-11) does not have enough time to fully set the EXOR gate. When phil is asserted, the EXOR output must still propagate through a pass transistor and two inverters, causing the counter to not be truly edge triggered. This problem becomes even worse in subsequent stages as the feedback signal must pass through an AND gate and the EXOR.

#### Testing of the Pass Shifter Network Chip

Three chips containing the pass transistor network (chip name: PASSNET) were fabricated and returned. Of the three, only one chip was found to be fully operational. On the other two, a 2 micron gap in the ground connection to the output pads prevented the pad outputs from becoming 0 volts. The pads would pull down only to 1V. Otherwise, these two chips operated as designed as well.

The functions tested on PASSNET are as follows: decoding of PLA, sign extension of output, and two's complementation

of output. In addition, the input and output pads were tested for operational limit. The initial testing was done at 5 kHz, but the chips were found to be operational to 13.8 MHz. The test output pad followed an input square wave reasonable well up to 17 MHz with about 1 volt attenuation in signal level.

Operation of the PASSNET chip at 13.8 MHz was encouraging in light of the individual operation of some of the basic cells. The two phase unit, which seemed to be at fault in basic cell testing seemed to be able to run ok with the smaller load placed upon it in this configuration. Also, the PLA, which was not speed tested alone, was shown to be useable at high speeds. At the same time, the overall design of the shifting network and associated logic was proven.

#### Testing of the Phase Accumulator Chip

Twenty six chips containing two versions of the phase accumulation circuitry were received. Thirteen chips (chip name PHIACCUM) contained the basic accumulator without the test configurable shift register outputs. The other thirteen (chip name PHIACCUM2) did have the shifter. The version without the shifter output was used to fully examine the internal nodes to determine the validity of the clocking scheme. The shift register version, which will be used in

the chip set, was fabricated apart from the shifterless version to provide more opportunity to fully test the accumulator.

Several DC tests were made on these chips. The clock and reset inputs were checked against their respective outputs and the complemented outputs. The non overlapping clock generators functioned correctly. The outputs were shown to be zeroed upon assertion of the reset signal.

Again, the two phase unit gave cause for concern. The separation was large, as experienced before, but also, the two phase unit with input ( $\phi_2$ .AND.reset) seemed to be time shifted such that its output was still high until  $\phi_1$  was asserted. The AND gate producing the product of the reset signal and  $\phi_2$  was suspect but this could not be shown. Also suspected of causing the large delays was the size of the load placed upon the extra large buffers on the output of the two phase unit. About 20 pass transistor gates were driven from each buffer, a few more than for what they had been designed.

The next DC test to be performed was to exercise the test configurable output registers as a shift register. The shift register tested as planned up to about 1.25 Mhz. It is believed that this limit is also the result of improper and

slowed timing caused by the extra test points which were placed on the clocking lines. Another flaw was a 2 micron routing gap which caused the MSB of the shift chain, as well as the MSB of the accumulator to be connected to the output pad in only three of thirteen (PHIACCUM2) chips. A similar error was committed on the input of the LSB of the PHIACCUM chip.

Once the peripheral circuits had been checked, the general operation of the accumulator was inspected. During testing of the accumulator, a fault was noticed in the input registers. Of twelve chips tested, only 1 of the 84 input latches worked as designed. The 83 defective latches would latch an input of 'zero', but once this was done, they would not store a 'one'. This seemed to indicate that the feedback transistor was being shorted out, causing a short to ground in the second inverter once the latches had received a 'zero' input. A serious fabrication error is suspected since the latch cell had been tested numerous times with no errors seen. The chip will be studied under a microscope to ascertain the full extend of the flaw.

The defect in the input latches prevented speed testing of the adder section itself. The first bit of the adder was found to be operational. This was done by flooding its input and the carry\_in with various inputs and observing correct

outputs from the adder. This procedure also revealed an error in the accumulator latches as well. Of 27 latches tested on three chips, none worked. Since the input and accumulation latches (which are both the same design and layout cell) have errors, it is assumed that an error in translating the cell description to fabrication mask was made. Again, it is hoped that the reason for this failure will be determined by observing the chip under a microscope.

The last measurement of these chips was another attempt to measure the delay through a pair of inverters. The signal to be measured was the output of the adder through the reset latch on the PHIACCUM chip. The second signal was expected to be delayed by only the two inverter delay, and by the charging of the capacitance of the feedback line which carried the sum signal back to the accumulator latch at the top of entire accumulator. Using the simplified transmission line equation given by Weste [19] ( $\text{delay} = rcl^2/2$ ), the delay should have been on the order of 10 nanoseconds for the two inverters and 5 ns for the diffusion area. By assuming that the pullup and pulldown transistors provide the circuit's only R, and the diffusion is the C, a delay of 40 ns is expected. Instead, a delay of 132 ns was measured. No reason has been found for this large change in the delay.

## Testing of the Coefficient Accumulator Chip

Because of the extensive test capabilities which were incorporated on the similar phase accumulator chips, only limited tests were designed into the coefficient accumulator system. These test ports, though, did allow for full testing of the chip.

As with the phase chips, the 'DC' tests were conducted first. The clocking circuitry was found to be functional on all 12 of the chips. Delays were similar to those noted on the phase chips. Again, the loading of extra test points is believed to be the cause. The output latch control circuitry, though, was found to be in error. An 'OR' gate input was stuck and thus would not allow the outputs to be viewed unless the reset signal was asserted. This, of course, precluded the full testing of the accumulator since the reset line caused 'zeros' to be loaded into the accumulator latches. This unit, as well as the phase system, will be examined further to determine the precise cause of the failures.

The adder unit itself, though, did seem to work correctly. This was noted by examination of the first few bits which showed correct correctness of the first three adder bits and of the first two carries. These tests also

showed that the input latches were, in fact, working. The accumulator latches also seemed to be working since the answers followed from what was to be expected if the accumulator latches were entering 'zeros'. (These latches, though, could have been stuck at 'zero' but no method was available to test this.) No timing information could be gleaned because of the faulty output control.

### Summary

An overview of the testing scheme which was used to test the Quantized Sinusoid DFT chips was given. Testing of the basic cells was described. All of the basic cells were found to be functional to the point where they could be used to operate the QSDFT algorithm at moderate speeds. The testing of the three chips in the chipset was also detailed. The pass transistor shift network chip was found to operational well above the expected operational limit. The phase accumulator chips were found to contain fabrication defects which prevent them from being used as is. The accumulating section of the coefficient accumulator chip was found to be functional except for a single gate which apparently has fabrication errors. The system control section also contained errors which prevented it from working.

## CHAPTER 6

### DISCUSSION OF RESULTS AND SUGGESTIONS FOR FUTURE RESEARCH

The design, fabrication, and testing of a Quantized Sinusoid DFT algorithm implemented in an nMOS VLSI technology have been described. The results of functional and operational testing of the hardware will now be summarized. In addition, suggestions for further research, design changes, and alternative uses will be considered.

#### Functional Results

In general, only a few of the chips were found to be fully functional. The cause of the errors were found to be mainly fabrication related. The design and layout errors which were found were not fatal to the operation of the chips as a chipset. The operational speeds of the chips which worked was found to be slower than the proposed 3 Mhz use. As noted, operation of some of the basic cells was not as good as was predicted from simulation and design.

These chips will now be consolidated on a single chip with further design considerations to provide tolerances for fabrication errors. To this end, simple changes such as redundant registers will have to be employed to allow for fabrication faults. In addition, probe points will be added (instead of using extra input/output pads) to allow access to the internal nodes of the circuitry.

Another lesson which was learned from this project is that the amount of time necessary to design and debug a completely custom control cell is often not worth the trouble. As was seen from the counter cell, small problems which may not appear during simulation can cause the cell's use to be limited. There were four fabrication runs for the counter. Of these runs, none of the counters could be operated above 3 Mhz. Simple flip-flop based designs can easily manage this speed without the design time penalty. In addition, there is no significant area savings with the custom since control cells usually do not occur with great frequency.

The above assertion, however, is not to say that small custom cells are not desirable. Instead, time should be spent mainly on those cells which are replicated many times or are to be used in an array fashion. In these instances, small cells will save area, and will allow for faster

operation due to decreased communication costs over the now smaller array area.

A final count of the project reveals that over 30 functional basic cells were designed and/or entered into the URI nMOS database. From these cells, approximately 10 general purpose modules were created. These 10 modules were then assembled into three chips containing some 2300 transistors.

### Operational Results

The supreme test of any hardware system is its ability to perform useful (and correct) calculations. Because of the fabrication errors, only the simulations of the chips' operation can be viewed to gain insight to the system's usefulness at this time. These simulations served not only to produce test vectors for exercising the circuitry, but to provide an example of the use of the chip.

As is seen in Figures 6-1 and 6-2, the Quantized Sinusoid DFT can be used to determine the principle components of a signal. Shown in Figure 6-1 is the QSDFT and regular DFT versions of a SINE Fourier Transform for a

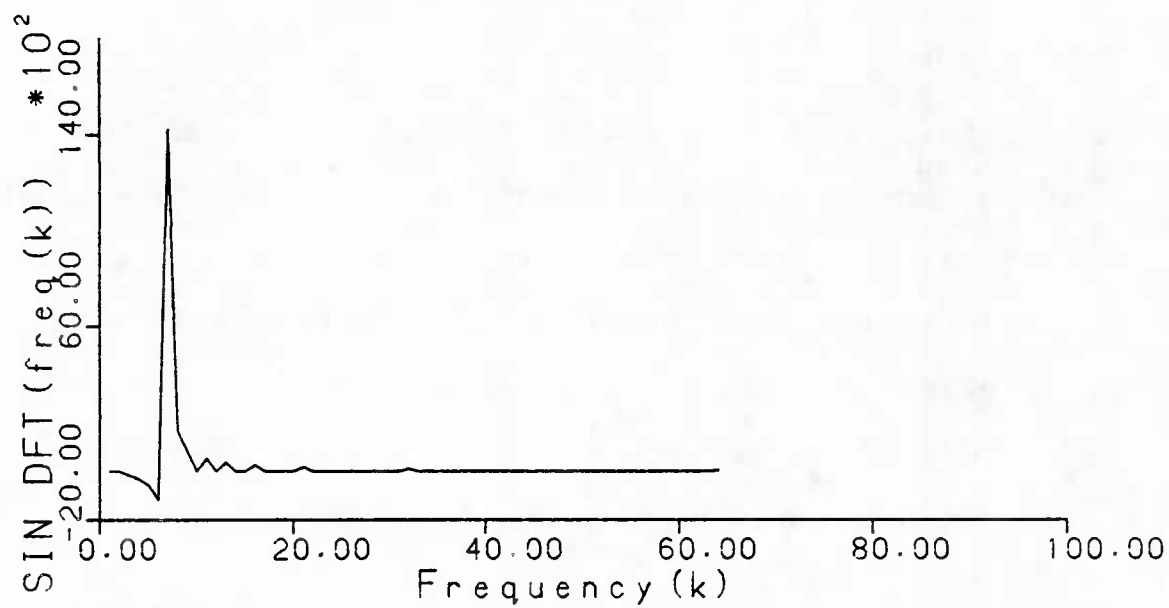
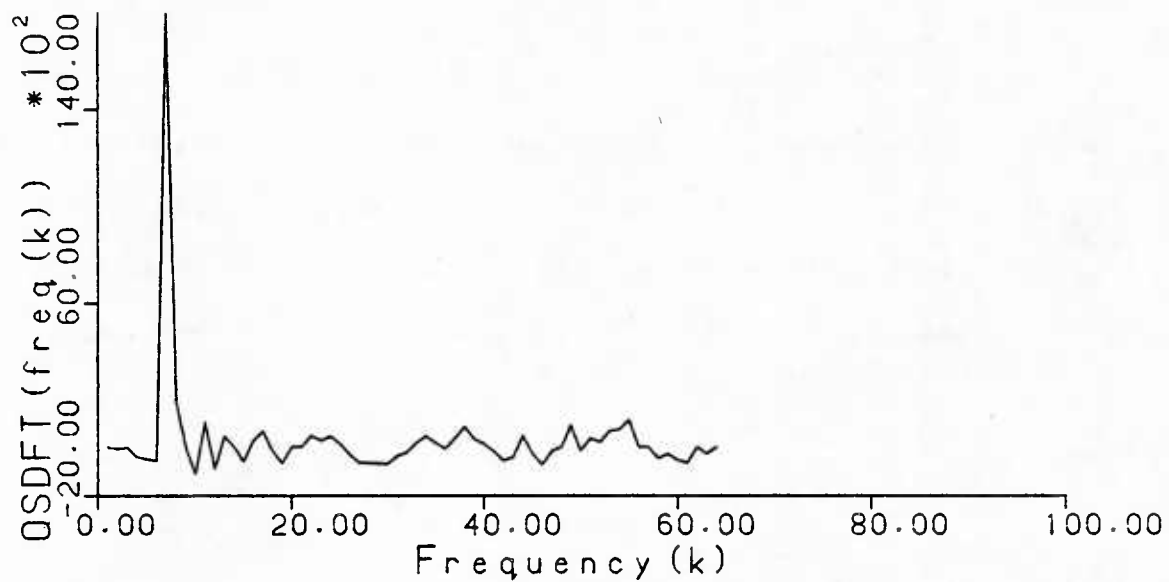


Figure 6-1

Comparison of SIN DFT and QSDFT (Single Component Sinusoid)

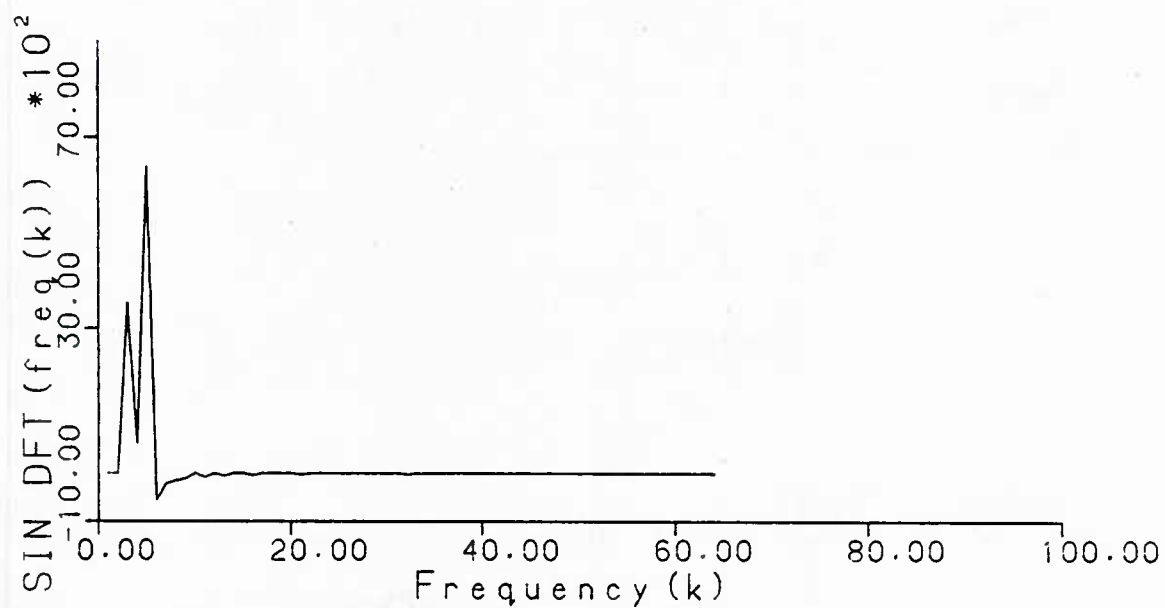
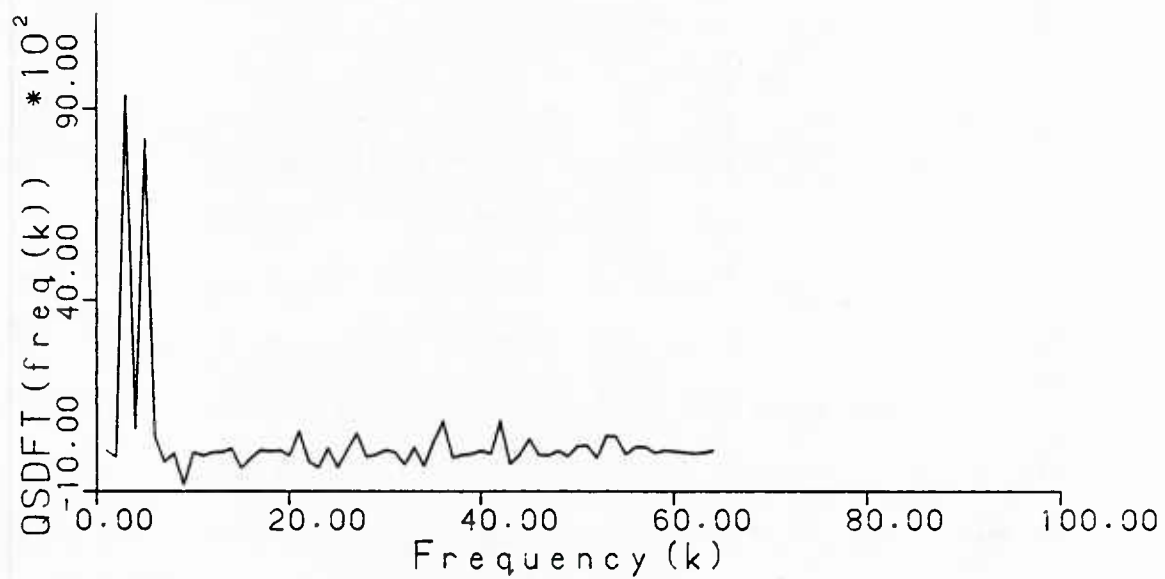


Figure 6-2

Comparison of SIN DFT and QSDFT (Double Component Sinusoid)

single component sinusoid  $\{y(n) = \sin(2\pi n/7)\}$ . Figure 6-2 shows results for a two component sinusoid where  $y(n) = \sin(2\pi n/3) + \sin(2\pi n/5)$ .

A general purpose computer, of course, could have calculated the same results in less time than the chip set would have done while being run as a co-processor for the VERSAbus. With input and output hardware though, the chip set can complete the calculation much faster than it would while in its testing mode. When other changes (as proposed below) are implemented, even faster and more versatile operation of the system will be possible. This exercise, however, has shown that this hardware implementation (sans fabrication errors) can be useful as a basis for building signal processing hardware.

#### Suggestions for Hardware Changes

As has been noted earlier, the general design of this chip allows for its use in several algorithms. To effect this use, it is necessary to make several hardware changes in order for the system to be utilized to its fullest potential.

The most obvious alteration is to include all of the circuitry on a single chip. It is to be expected that both the sine and cosine portions may be implemented on a single silicon die. A generalized floorplan of this arrangement is shown in Figure 6-3. This will require at least a 64 pin dual in-line package or possibly a small pin grid array due to the large parallel output requirements. A smaller pin out package may be used with a corresponding penalty for output bandwidth.

Another major change would be the increase of input data size to 12 bits. This would allow for the use of the more popular 12 bit A/D converters. Since the architecture is mostly bit slice in nature, only the shifter network will require major rework.

A change in implementation technology from nMOS to cMOS will provide several advantages for the system. The primary reason for using cMOS is a diminished power requirement. This will reduce the power needs for low frequency operation such as when the system is used for audio signals, but will not lessen the power requirement for the more dynamic high speed operation modes. Should, however, mass storage be implemented onboard, cMOS will provide a significant power savings over nMOS. A cMOS implementation will also for the use of design features of the currently available MOSIS cMOS

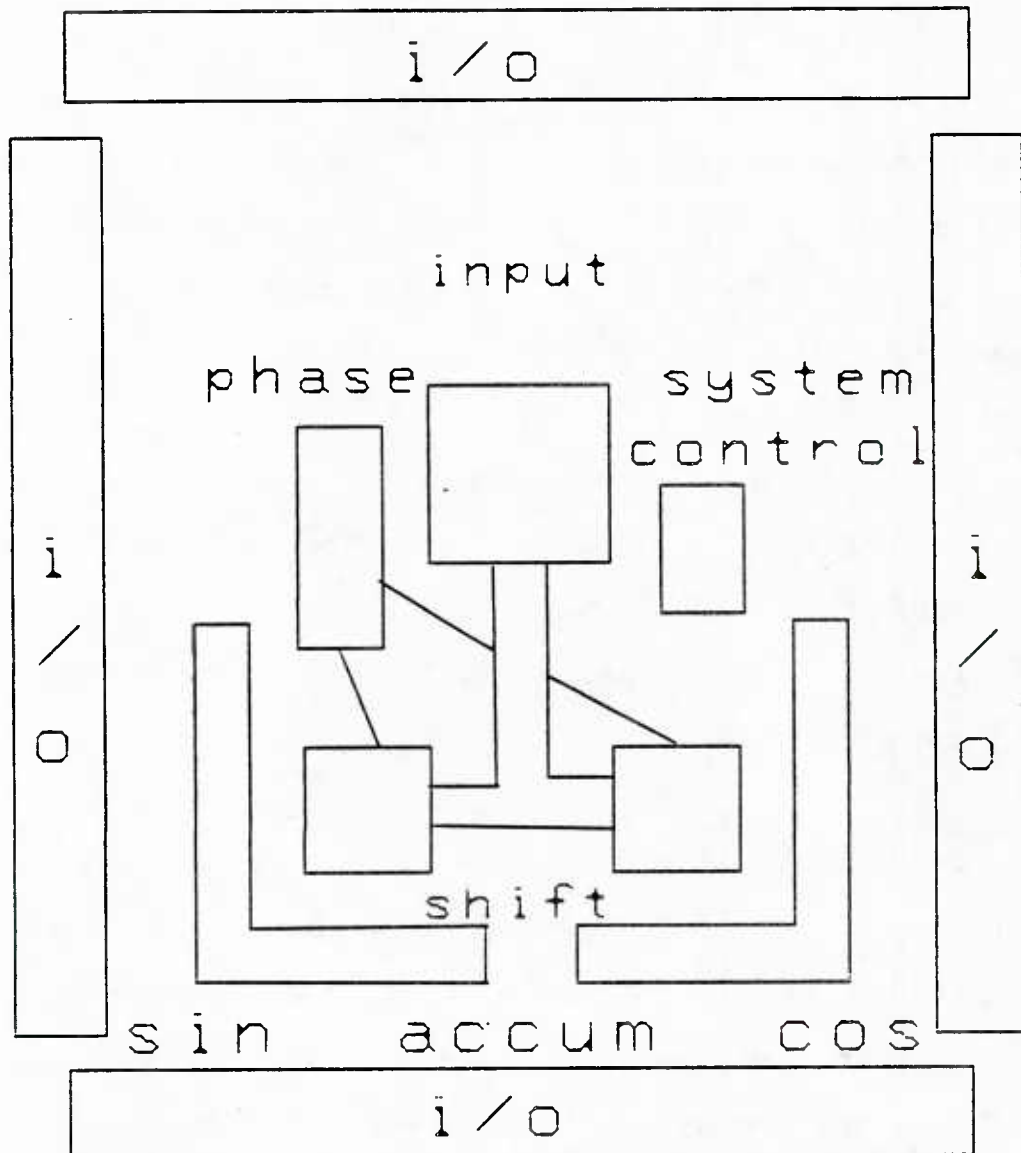


Figure 6-3  
Proposed Final Hardware Floorplan

processes such as second metal for routing and interconnects, and the ability to scale down a given design without having to redesign the system.

Internally, a few changes are desirable. A faster system clock would be the main goal of these changes. A carry lookahead type adder is needed to reduce what is now the limiting factor of the system. With a 20 bit addition time of well under 50 ns, the system clock could be increased to 20 MHz. This would allow for 16 point DFT calculations to be made above 1 MHz. Cascading of systems would further increase this throughput. With a faster adder unit for the phase accumulator, an input cache may be needed so as to not disrupt the pipelined operation flow. Additional control circuitry may be needed to provide for switching between caches if a dual input cache is implemented as proposed previously. Also, the decode logic may have to be hardwired, instead of implemented using a PLA since the PLA may not be fast enough for the 20 MHz clock rate.

Finally, more controls will have to be added to allow for such things as input data type selection. If internal storage is to be added, as was proposed in the original QSDFT algorithms, controls for its use will also be needed. In general, complete control over the entire chip will have

to be provided to allow the system to reach its full range of versatility.

#### Suggestions for Other Uses of the Hardware

With the above improvements, the versatility of this architecture will be increased. Keeping these improvements in mind, several other operations in which this system may be used will now be discussed. Although not inclusive, the following list of uses should convey a sense of the possible uses for this architecture.

Another use which involves the use of the system to calculate an estimated frequency component is the low pass filter [13]. In this use, the time domain signal is multiplied by  $\text{EXP}[j\omega_c t]$ . This shifts the signal centered around  $\omega_c$  to the origin. The signal can then be filtered with a low pass window such as is shown in Figure 6-4. Using eight divisions of the quantized sinusoid (as was done in the QSDFT algorithm) the shifted time domain signal may be accumulated into the appropriate bin. These bins may then be shifted and accumulated again for each position of the secondary quantized sinusoid.

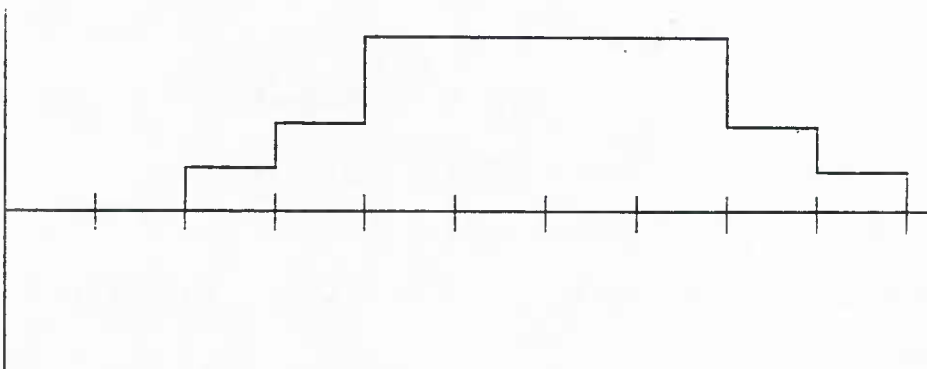
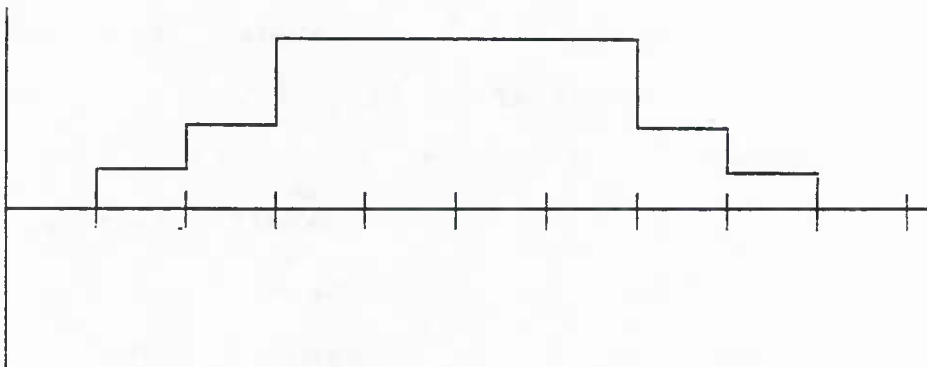
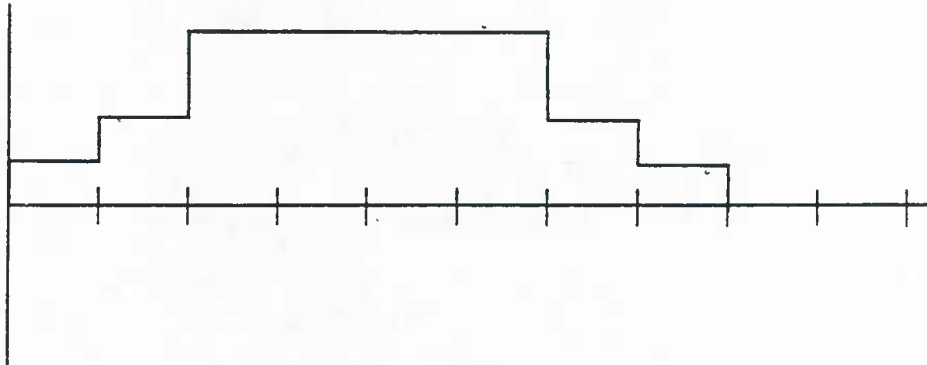


Figure 6-4  
Low Pass Filtering Windows

To simplify hardware though, a second QSDFT chip may be used to perform the bin calculations assuming the output from the first QSDFT chip has been scaled or the input to the secondary QSDFT chip is enlarged. This method has the added advantage of allowing adaptive filtering since both the original  $w_c$  and the secondary low pass 'frequency' may be altered on the fly without having to reset the entire system.

Another possible use of the system involves Permuted Difference Coefficients PDC's [20]. In this algorithm, the coefficients of an FIR filter are reordered according to size. The reordered coefficients are differentiated creating the need to integrate the input data. The differentiation of the coefficients causes the elements of this vector to diminish in size, or to possibly introduce zero valued elements. After several passes through the algorithm, only a few non-zero values remain in the coefficient vector. Thus, the dot product of the newly integrated input data vector and the differentiated coefficient vector (with its many zero values) will contain few multiplies which actually need to be calculated.

The QSDFT hardware may be of use to this algorithm if a reverse method of determining the permuted coefficient differences can be found. In this case, the somewhat rigid

hardware gains flexibility from the ability to input control values (i.e. the phase increments of the QSDFT algorithm). A method which would work the PDC algorithm backwards might possibly allow for the use of the existing QSDFT hardware to perform the calculations.

### Conclusions

An nMOS VLSI system which may serve as the basis for several signal processing algorithms has been described. At present, it is implemented as a three chip set. The design's use to calculate the sine portion of a 64 point DFT has been shown. Various other algorithms which can be adapted to use the system have been discussed. Possible modifications to the hardware which will increase the flexibility of the system were given.

## REFERENCES

- [ 1 ] Kung, Sun-Yuan, 1982, Impact of VLSI on Modern Signal Processing, Proc. USC Workshop on VLSI and Modern Signal Processing, pp. 123-132
  
- [ 2 ] Owen, Robert E., 1984, VLSI Architectures for Digital Signal Processing, VLSI DESIGN Vol. 5, No. 6, pp. 20-28
  
- [ 3 ] Swartzlander, Earl E. and Hallnor, George, 1984, Fast Transform Processor Implementation, Proc ICASSP, pp. 25A.5.1
  
- [ 4 ] Cooley, J.W., and Tukey, J.W., 1965, An Algorithm for the Machine Computation of Complex Fourier Series, Math. Computation, Vol. 19, pp. 297-301
  
- [ 5 ] Cooley, J.W., Lewis, P.A.W., and Welch, P.D., 1967, Historical Notes on The Fast Fourier Transform, IEEE Trans. Audio Electroacoust., Vol. AU-15, pp. 45-55
  
- [ 6 ] Oppenheim, Alan V. and Schafer, Ronald W., 1975, Digital Signal Processing, Prentice-Hall

- [ 7 ] Goertzel, G., 1958, An Algorithm for the Evaluation of Finite Trigonometric Series, Amer. Math. Monthly, Vol. 65, pp. 34-35
- [ 8 ] Rabiner, L.R., Schafer, R.W., and Rader, C.M., 1969, The Chirp z-Transform Algorithm, IEEE Trans. Audio Electroacoust., Vol. AU-17, pp. 86-92
- [ 9 ] Runge, C., 1903, Zeit. fur Math. and Physik, Vol. 48, pp. 443
- [ 10 ] Runge, C., 1905, Zeit. fur Math. and Physik, Vol. 53, pp. 117
- [ 11 ] Danielson, G.C. and Lanczos, C., 1942, Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering From Liquids, J. Franklin Inst., Vol. 233, pp. 365-380 and 435-452
- [ 12 ] Runge, C. and Konig, H., 1924, Die Grundlehren der Mathematischen Wissenschaften, Vorleungen uber Numerisches Rechnen, Vol 11. Berlin: Julius Springer

[ 13 ] Tufts, D.W., Rorabacher, D.W., and Moser, W.E., 1970, Designing Simple Effective Digital Filters, IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 2

[ 14 ] Tufts, D.W., and Melissinos, C.D., 1984, Simple, Effective Computation of Principle Eigenvectors and their Eigenvalues and Application to High-Resolution Estimation of Frequencies, Submitted to IEEE Trans. on Acoust., Speech, and Signal Processing

[ 15 ] Mead, C. and Conway, L., 1980, Introduction to VLSI Systems, Addison Wesley

[ 16 ] Borsay, D., Collier, L., Edwards, R., and Wilkinson, J., 1983, CHIP-Circuit Hierarchy Integration Program, Proc. DECUS, Las Vegas, NV, October

[ 17 ] McCormick, Steven P., 1982, The Evolution of a Thirty-Two Bit Full Adder, Advanced Topics in VLSI Systems, 6.372, May

[ 18 ] Newkirk, John, and Mathews, Robert, 1983, VLSI Designer's Library, Addison-Wesley

[ 19 ] Weste, Neil N. E., and Eshraghian, Kamran, 1985, Principles of CMOS VLSI Design, Addison Wesley

[ 20 ] Nakayama, Kenji, 1982, Permuted Difference Coefficient Realization of FIR Digital Filters, IEEE Trans. Accustics, Speech, and Signal Proc., Vol. ASSP-30, No. 2, pp. 269-278

## BIBLIOGRAPHY

### VLSI Testing

Barrow, Harry, 1984, Proving the Correctness of Digital Hardware Designs, VLSI Design, Vol. 5, No. 7, pp. 64-77

Beucler, Frederick P. and Manner, Michael J., 1984, HILDO: The Highly Integrated Logic Device Observer, VLSI DESIGN, Vol 5, No. 6, pp. 88-96

Breuer, M.A. (ed), 1976, Diagnosis and Reliable Design of Digital Systems, Computer Science Press

Eichelberger, E.B., 1973, Method of Level Sensitive Testing a Functional Logic System, US Patent 3761695

Frohwerk, R.A., 1977, Signature Analysis: A New Digital Field Service Method, Hewlett Packard Journal, May, pp. 2-8

Gannett, Joel W., 1984, VLSI Design For Testability, VLSI Design and Architecture, (N.G. Einspruch ed.) Academic Press N.Y.

Jack, M.A., 1983, Design for Testability, Semi-Custom IC Design and VLSI, IEEE, pp. 168-185

Mueldorf, E.I. and Savkar, A.D., 1981, LSI Logic Testing - An Overview, IEEE Trans. Comput., Vol. C-30, pp. 1-17

Nadig, H.J., 1977, Signature Analysis - Concepts, Examples, and Guidelines, Hewlett Packard Journal, May, pp. 15-21

Williams, T.W. and Parker, K.P., 1979, Testing Logics Networks and Design for Testability, Computer, Vol. 12, No. 10, pp. 9-21

Williams, T.W. and Parker, K.P., 1982, Design For Testability - A Survey, IEEE Trans. Comput., Vol. C-31, pp.2-15

## DISTRIBUTION LIST

	Copies
Statistics and Probability Program (Code 411(SP)) Office of Naval Research Arlington, VA 22217 Attn: Dr. Neil Gerr	3
Defense Technical Information Center Cameron Station Alexandria, VA 22314	6
Arthur J. Forrester ONR Resident Representative Harvard University Vanserg Building 29 Francis Avenue Cambridge, MA 02138	1
Commanding Officer Office of Naval Research 1030 East Green Street Pasadena, CA 91101 Attn: Dr. Clifford Lau	1
U.S. ONR Liaison Office - Far East Attn: Scientific Director APO San Francisco 96503	1
Navy Library National Space Technology Laboratory Attn: Navy Librarian Bay St. Louis, MS 39522	1
U.S. Army Research Office P.O. Box 12211 Attn: Dr. J. Chandra and Dr. Sander Research Triangle Park, NC 27706	1
Director National Security Agency Attn: R51, Dr. Maar Fort Meade, MD 20755	1
ATAA-SL, Library U.S. Army TRADOC Systems Analysis Activity Department of the Army White Sands Missile Range, NM 88002	1
Library, Code 1424 Naval Postgraduate School Monterey, CA 93940	1
Technical Information Division Naval Research Laboratory Washington, DC 20375	1

	Copies
Mr. James Smith Code 340R Naval Air Systems Command Washington, DC 20361	1
Naval Coastal Systems Center Code 741 Attn: Mr. C.M. Bennett Panama City, FL 32401	1
Dr. John Schuster Space and Naval Warfare Systems Command Code 612 Washington, D.C. 20363-5100	1
Mr. Nicholas Smea Space and Naval Warfare Systems Command Code 612 Washington, D.C. 20363-5100	1
Dr. Steven Sacks Space and Naval Warfare Systems Command Code 612 Washington, D.C. 20363-5100	1
Library Naval Ocean Systems Center San Diego, CA 92152	1

U224248